LIFE CYCLE COST IMPROVEMENTS FROM FOD ELIMINATION ON TURBOFAN COMPRESSOR ROTOR BLADES

A Thesis
Presented to
The Academic Faculty

by

Mauricio J. Guadamuz

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
August of 2016

LIFE CYCLE COST IMPROVEMENTS FROM FOD ELIMINATION ON TURBOFAN COMPRESSOR ROTOR BLADES

Approved by:

Dr. Dimitri N. Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. George Kardomateas
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Kelly Griendling
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Daniel P. Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Joseph Bobinis
LM Fellow Senior, SoS
Lockheed Martin Corporation

Date Approved: July 20, 2016

To Mauricio, Georgia, Lia and Gianna

ACKNOWLEDGEMENTS

I would like to thank my lord and savior Jesus Christ for being the catalyst and heading for my journey. I would like to thank my family for supporting/withstanding my schedule and focus – without their support this endeavor would not have been as rewarding or valuable. I would like to thank my friends and colleagues for their encouragement, availability and coaching – without those shoulders to stand on, it would have been harder to see further.

I would like to thank my advisor Prof. Dimitri Mavris for his academic, technical and personal advice – without his kind and empathetic advice I might have lost the balance needed to be a parent, professional and Ph.D. student. I would like to thank Dr. Ray O Johnson for believing in me and for championing my return to Georgia Tech to complete my Ph.D. I would like to thank Manuel (Manny) Sanchez and Joseph Bobinis for their coaching and deep insight into the world of complex systems.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS & SYMBOLS

C_p	Specific heats
C_Θ	Whirl or absolute tangential velocity
d	Runway width-wise FOD particle pitch
D	Diameter
D_l	Engine inlet diameter
DRPT	Debris removal process and/or technology
E	Material Young's Modulus
FOD	Foreign Object Debris or Damage
g	Gravitational acceleration
h	Engine ground clearance or ground to engine highlight clearance
H_s	Static enthalpy
H_t	Total or stagnation enthalpy
IGV	Inlet Guide Vane
IPT	Inspection process and/or technology
L	Length
M_oER	Measure of Eradication
M_oR	Measure of Robustness
N	Runway width space counter between FOD particles or engine spool speed
P	Pressure
r	Radial distance from engine centerline axis
RMT	Robustness method and/or technology

T_s	Static temperature
T_t	Total or stagnation temperature
W	Width
w_{ER}	Weight of Eradication
w_R	Weight of Robustness
W_Θ	Relative tangential velocity
z	Height from ground datum
η	Efficacy
λ	Twist or stagger angle
ρ	Density
σ	Stress or blade row solidity
ω	Angular velocity

SUMMARY

Currently, Original Equipment Manufacturers (OEMs) and operators of military aircraft with low-bypass jet engines do not consider the expensive and possibly catastrophic consequences of Foreign Object Damage (FOD) to the High Pressure Compressor (HPC) rotor blades due to impacts from Foreign Object Debris (FOD, acronym context interchangeable with damage) during runway operations early in the takeoff phase in a systematic/symbiotic practice – 1st level approximations to the affect/effect of FOD to these critical, rotating structures are needed during early design studies of conceptual military jet aircraft – equally important is the need by operators of current aircraft to be able to gage the effectiveness and affordability of their operational paradigm (baseline) and of enhanced configurations. The presence, distribution and nature of FOD on the runway are analyzed separately from its structural or economic degradation to turbomachinery of the engine. Missing in the public domain is a symbiosis of the physics and probabilistic models for how FOD on the runway is aspirated via ground vortex action, how and where it is then ingested by the engine at the fan face, what trajectory through the turbomachinery does it follow to impact the rotor blades, where and at what velocities do impacts occur and ultimately what is the ensuing damage and its effect on crack growth at the location of the impact sites. Also missing in the public domain is a synergistic assessment of the Life Cycle Cost (LCC) impact of the current methodologies for mitigating the deleterious effects from FOD on the runway – equally needed is a methodology to gage the LCC implications of improved or new processes, technologies and design improvements to mitigate the effects of FOD on current and future engine/engine on aircraft designs.

A research and business case for the elimination of the FOD-exacerbated High Cycle Fatigue (HCF) failure mode caused by Hard Body debris impact damage on blades on the 1st rotor disk of the HPC of a modeled low bypass, turbofan engine of a military jet during runway operations early during takeoff will be made. The research objective of this thesis will be to determine if FOD elimination, the minimization of the FOD-exacerbated HCF failure mode that leads to 1st HPC rotor blade replacements and/or catastrophic engine losses while reducing or marginally increasing LCCs, is achievable and if so by what mix of technologies, processes and/or design changes at the primary and support systems. A detailed discussion on a proposed methodology and process will be presented along with a numerical/analytical experimental plan and questions to systematically test if FOD elimination can be achieved affordably in a LCC context.

To investigate how FOD-induced damage to the HPC rotor can be eliminated models were developed to estimate/calculate:

- Conditions for debris aspiration from ground
- Density and distribution of FOD particles on runway
- Probability that the ground core of the vortex line that sheds from engine fan face will move over a FOD particle
- Kinetics of FOD particle travel as it is aspired via ground vortex action to engine fan face – this regime occurs when aircraft engages engine to 100% power and begins to travel down runway for takeoff, commonly referred to as “High Suck, Low Speed” regime – flow outside of the engine is well below Mach 0.3 during this early takeoff phase and the condition of incompressibility is assumed

- In-engine flow total and static properties – the compressible flow regime through fan, stator, inlet guide vane up to first rotor disk – model also enables sizing of turbomachinery at specific runway conditions
- Kinetics of FOD particle travel as it is ingested through fan, stator, inlet guide vane up to first rotor disk
- Impact law for FOD particle and rotor blade
- Map of impact locations on rotor blade
- Centrifugally-induced, tensile, principal stress at impact site
- Statistical relationship between FOD particle velocity and depth/width of penetration at impact site
- Geometric factors needed in stress intensity calculations to fully determine state of stress at the tips of cracks that grow naturally from Equivalent Initial Flaw Size (EIFS) and/or FOD particle impacts
- Crack growth with and without surface treatments such as Laser Shock Peening that induce beneficial, residual compressive stresses to the rotor blades
- Current and improved runway FOD inspection and removal process performance

A methodology was developed to account for the LCCs associated with current and improved runway FOD inspection and removal as well as technology infusions such as automated debris detection, Damage Tolerance based interval engine inspection schedules and imparting structural robustness to rotor blades via surface treatments.

A total of 2,162 numerical/analytical experiments for FOD particle impact were performed (1,081 experiments for mild steel particles of 1.33 and 3.2 mm diameter) – revealing:

- Particle size matters – 3.2 mm diameter particles struck blades on the 1st HPC rotor less often than 1.33 mm diameter particles (20.8 % vs. 37.2% impact occurrence respectively)
- For the two particle sizes considered, 1.33 and 3.2 mm diameter, impact locations are distributed radially and chord-wise on the airfoil suction and pressure sides – debunking for these two particle sizes the trend of impact testing only at leading edges

For crack growth and LCC estimation a total of 19,392 numerical/analytical experiments, each representing an engine lifetime, were performed (9,696 experiments for crack growth and LCC estimation if impacts occurred or not for mild steel particles of 1.33 and 3.2 mm diameter) – revealing:

- Particle size matters – 3.2 mm diameter particles struck blades on the 1st HPC rotor less often than 1.33 mm diameter but have a significantly larger impact on LCCs, blade Replacements and Repairs - Vs. the current paradigm of two visual inspections and one runway sweep for debris on the runway for the larger particle, the technology infused configuration of Automated runway inspections and interval engine inspections has 30% higher LCCs but 92% fewer LRU Replacements
- Automation is key – automated inspections along with Laser Shock Peening (LSP) and interval engine inspections reduce significantly the risk of engine

catastrophic damage because flaws that would have gone critical are detected and repaired – allowing for a fuller realization of engine operation

- LSP has a small footprint on LCCs but a significant benefit in reducing blade replacements and repairs

1 INTRODUCTION/LITERATURE SURVEY

1.1 "IT'S AFFORDABILITY, STUPID"

The famous mantra coined by James Carville during the 1992 presidential campaign, "It's the economy, stupid" [1] is a sobering reminder of how the economy impacts key decisions: Bill Clinton beat a strong opponent on a message centered on national economics. Foreseeably for aerospace OEMs and operators the mantra needs to be "It's affordability, stupid". Affordability is defined by the International Council on Systems Engineering (INCOSE) and the National Defense Industry Association (NDIA):

- INCOSE definition – "Affordability is the balance of system performance, cost and schedule constraints over the system life while satisfying mission needs in concert with strategic investment and organizational needs" [2].
- NDIA definition – "Affordability is the practice of ensuring program success through the balancing of system performance (KPPs), total ownership cost, and schedule constraints while satisfying mission needs in concert with long-range investment, and force structure plans of the DOD" [3]. KPPs are Key Performance Parameters.

1.1.1 NEED FOR AFFORDABILITY IN MILITARY AVIATION

Defense budgets for the foreseeable future are on the decline [4, 5]; military acquisitions are already recalibrating to this new reality [5]. Since the height of defense spending in 1944 of 43% of U.S. GDP to 5% in 2012 (see Figure 1-1), national defense spending has been on a steady decline that is forecasted to continue [4, 6]. The Budget Control Act (BCA) of 2011 mandates defense spending to decline by $487 billion (in 2011 U.S. dollars) from FY 2012 through 2021 [7]. Current military programs are being reorganized to reduce costs and future programs will have to accurately forecast significant LCC improvements throughout program life early in the Requirements Definition and

Conceptual Design phases [8, 9, 10]. In the past decade the F-35 Lightning II Program has seen significant rebaselining and cost scrutiny to set it back on track to acceptable per unit acquisition costs: rescheduled/rescaled deliveries, implementation of more stringent program cost/managerial controls and capability rebaselining [8]. The Weapons Systems Acquisition Reform Act of 2009 puts Systems Engineering (SE) practices to law by driving improved design-to requirements, life cycle plans and financial management early in the acquisition process [11]; the implications of this law change the focus of government acquisition to affordability. Senator Carl Levin, one of two proponents of the law, contextualized its need and implications to the Senate Armed Services Committee: "The key to successful acquisition programs is getting things right from the start with sound systems engineering, cost estimating, and developmental testing early in the program cycle. The bill that we are introducing today will require the Department of Defense to take the steps needed to put major defense acquisition programs on a sound footing from the outset. If these changes are successfully implemented, they should help our acquisition programs avoid future cost overruns, schedule delays, and performance problems" [12]. Senator John McCain, the ranking member of the Armed Services Committee and the second proponent of the law, echoed the need for the legislation: "The Weapon System Acquisition Reform Act of 2009 is an important step in efforts to reform the defense acquisition process. This legislation is needed to focus acquisition and procurement on emphasizing systems engineering; more effective upfront planning and management of technology risk; and growing the acquisition workforce to meet program objectives" [12].The fear of the "Death Spiral" in aerospace defense is here to stay: acquisition unit costs of new defense-centric aircraft are going up, driving fewer acquisitions of more expensive to maintain aircraft, which in turn drive unit acquisition costs up even further [13, 14].

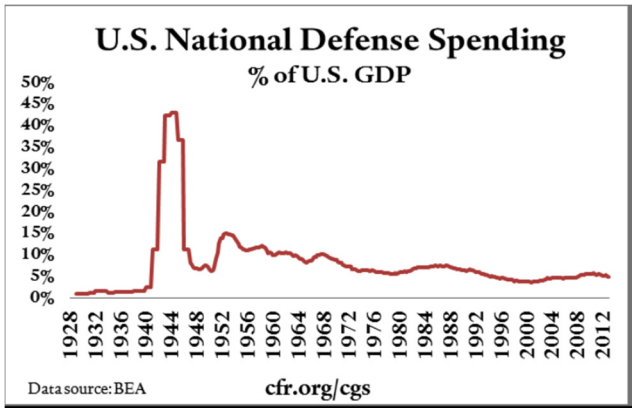


Figure 1.1: US National Defense Spending

Operations and Support (O&S) of military, fixed wing aircraft account for 63% of LCCs – half of the O&S LCCs are attributable to Maintenance, Repair and Overhaul (MRO) on jet engines [15]. HCF, a phenomenon where blade material is weakened by low stress cycles repeated many times (typically begins in the range of 10^5 to 10^7 cycles), is the prime driver of component failure and associated costs in modern jet engines of military aircraft and is a significant concern for future engines – FOD catalyzes HFC failures [16]. Large data sets for bird and debris strikes on engine components are not readily available for military aircraft as they are for their commercial counterparts, but public domain evidence suggests direct costs to mitigate the effects of FOD are comparable – a strong indicator that on the military side FOD is pervasive and expensive to deal with [17]. Runways by far pose the greatest location risk associated with FOD to jet engines – damage causing debris strikes are more frequent and more expensive than bird strikes to jet engines during aircraft runway operations at takeoff. In the context of this thesis runway operations are synonymous with and focused on the early phase of takeoff.

Quantifying the costs associated with FOD for current, improved or future military aircraft engine designs may be a prime opportunity to assess and predict improvements to LCCs – a key requirement that already makes or breaks an acquisition decision – with the ability to estimate early during design studies, even if only via 1st level approximations, an OEM would have a competitive edge on the competition by proposing engines/engines in

aircraft configurations and improved operating procedures that would empower the operators with increased robustness and affordability. An aggressive, novel approach must be implemented that delivers improved life cycle affordability or the self-fulfilling prophesy of fewer acquisitions of ever more expensive to maintain aircraft becomes a spiraling reality.

1.1.2 Need for Affordability in Commercial Aviation

In the commercial aviation sector, affordability, and its beneficiary, profitability have seen volatile swings since deregulation of the industry in the United States in 1978 [18, 19]. Deregulation allowed the United States government to transfer, via the Airline Deregulation Act of 1978, its management responsibilities over the airline industry to private firms; no longer would the government regulate ticket prices, services and growth of airline networks [19, 20]. America's leap into deregulation of its airlines was quickly seen as a success by most of the world; deregulation in foreign markets followed suit largely because governments saw benefits passed onto their travelers (lower fares, increased airline choice, better service) [18, 20]. Deregulation drove stiff competition and lower fares amongst airlines, leaving very little, if any margins; airlines are expected to profit on average $5.94 per departing passenger in 2014, fares are 40% less than what they were in 1979 [21, 22] (see Figure 1-2 and Figure 1-3). Despite continued passenger growth and record ridership profit margins in commercial aviation are fleeting; according to the International Air Transport Association (IATA) the average net profit margin for airlines globally was 0.0% (1990 – 2012) [23, 24].

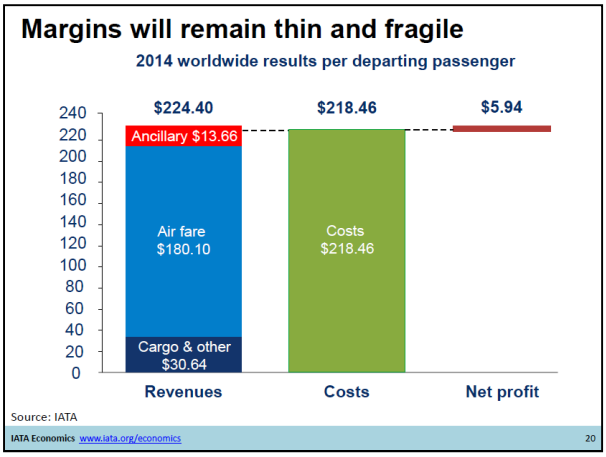


Figure 1.2: Revenues Costs Net Profit per Departing Passenger

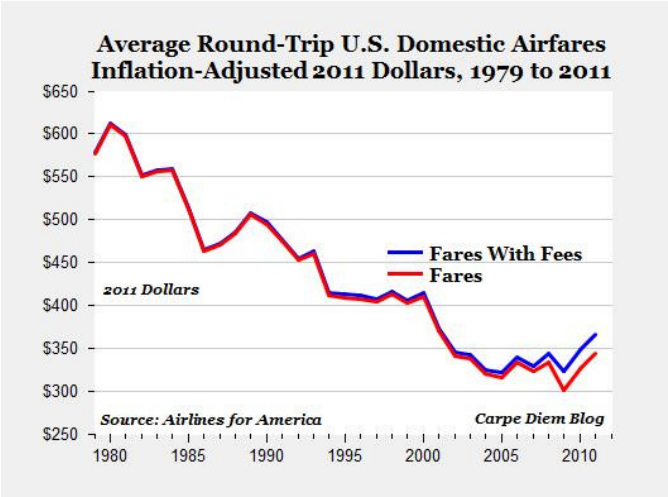


Figure 1.3: Average Round-Trip U.S. Domestic Airfares

Higher costs of fuel burden heavily on affordability for this industry; currently fuel takes up the largest share of an airline's operating costs at over 30% [23, 25]. Airlines are hamstrung on fuel price, they cannot control it; airlines must continue to aggressively improve their cost efficiency in non-fuel operations – maintenance accounts for 11% of operating expenses, it may provide a sizeable, actionable room for affordability improvements [28, 29] (see Figure 1-4). Novel approaches to cost reduction must be proposed to drive improved affordability in this industry.

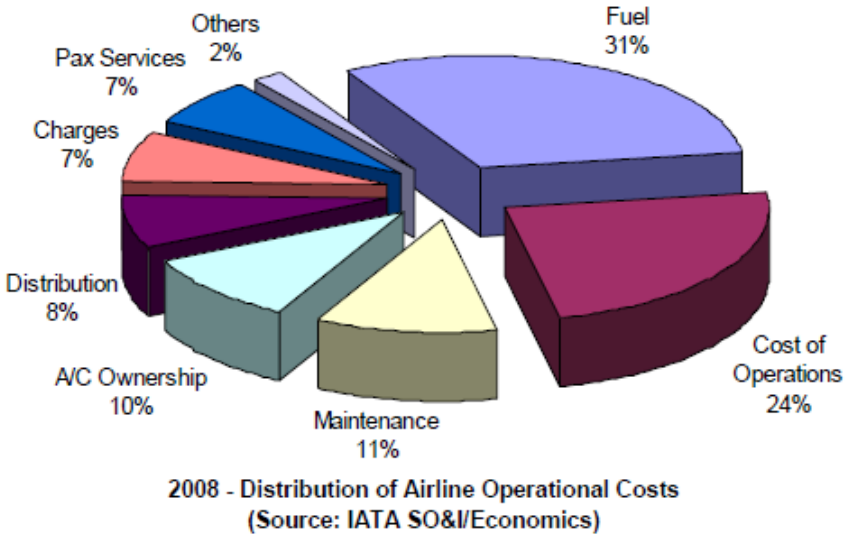


Figure 1.4: Airline Operating Costs

FOD costs the global airline industry $14.86 billion (in 2013 U.S. dollars) PA in direct and indirect costs; $1.35 billion (in 2013 U.S. dollars) PA are direct costs that are burdened with a 10X multiplier from indirect sources such as the forty-nine considered by McCreary [30]. The world's airlines netted large profits in 2013 of $12.9 billion with total revenues of $708 billion, their Total Operating Costs (TOCs) are $695.1 with 2.14% of these burdened just by FOD – FOD still outpaced them financially and hardly anyone noticed [31]. FOD strikes occur twice as often as bird strikes and are more likely to cause damage – whereas the current paradigm for airworthiness certification mandates structural robustness towards bird strikes [30]. For total aircraft MRO, 35% is attributable to the engines – making the single largest cost contributor for total MRO on commercial aircraft [32].

1.2 DEFINITIONS, COSTS AND CONSEQUENCES OF FOD

1.2.1 WHAT IS FOD?

FOD in the context of this thesis is best defined by the National Aerospace FOD Prevention, Inc. (NAFPI) [33]:

Foreign Object Debris (FOD): “A substance, debris or article alien to a vehicle or system which would potentially cause damage”.

Foreign Object Damage (FOD): “Any damage attributed to a foreign object that can be expressed in physical or economic terms which may or may not degrade the product’s required safety and/or performance characteristics”.

1.2.1.1 FOD is a Disabling System

Systems Engineering (SE) as defined by Blanchard and Fabrycky is [34]: “the application of efforts necessary to:

1. transform an operational need into a description of system performance parameters and a preferred system configuration through the use of an iterative process of functional analysis, synthesis, optimization, definition, design, test, and evaluation;
2. integrate related technical parameters and assure the compatibility of all physical, functional, and program interfaces in a manner that optimizes the total system definition and design; and
3. integrate performance, producibility, reliability, maintainability, manability, supportability, and other specialities into the total engineering effort.”

In the tradition of Systems Engineering a Primary System is defined by Bobinis as: “The product design, output mechanisms for generating the functions the system was designed for” [35]. No Primary System is perfect; all primary systems require Enabling Systems that provide countermeasures in the form of Enabling Functions to overcome design deficiencies. Bobinis contextualizes the relationship of Enabling Systems and Enabling Functions to the Primary System (the airplane in the context of this thesis) as:

"Enabling functions countermeasure deficiencies in design, in terms of unreliability and quality of output, in the generation of primary system functions. Enabling Systems are often referred to as Support Systems". Because FOD, as defined by NAFPI [33], offers no enabling functional value to the Primary System it is a Disabling System – it exists only to degrade the performance of the Primary System. It will be shown later in this document that current FOD "management" strategy is least affordable with external Support Systems – FOD Elimination internal to the Primary System will drive greater life cycle affordability.

1.2.1.2 FOD Classifications

Before FOD causes physical damage to a system it is called Foreign Object Debris. Once debris causes physical detriment to a system it is called Foreign Object Damage. Both classifications of FOD are found at commercial and military operation of aircraft [17, 30, 33, 36]. Runway operations are at the mercy of both categories of FOD.

Foreign Object Debris:

Foreign Object Debris comes in two major designations: Soft Body FOD and Hard Body FOD [17]. Soft Body FOD includes, but is not limited to: ice, birds and other animals, leaves, paper, cloth rags, wood, plastic. Hard Body FOD includes, but is not limited to: personal equipment, tools, rocks, pebbles, dust, metallic debris and small parts left on runway or tarmac from other aircraft. The difference between Soft Body and Hard Body FOD arises from the type of damage they may cause [17]:

- Soft Body FOD in the form of birds and ice will cause soft impacts, as relative to the aircraft in flight these types of FOD are moving slowly – these soft, slow impacts are able to transfer most of their momentum to the structure they impact with ensuing damage over a large area with large plastic deformations, but no

material loss, tearing or cracking – this type of damage is typically constrained to flight and runway operations

- Hard Body FOD in the form of small rocks, dust, runway material, small parts will cause hard impacts whose damage is very localized, largely in the form of material loss, tearing and cracking – this type of damage typically occurs on runway operations

Foreign Object Damage:

Foreign Object Damage resulting from Foreign Object Debris can take the form of, but is not limited to loss of function, decreased efficiency, material loss, tearing, cracking, plastic deformation and electronic systems degradation [17, 33, 37]. The economic impact of Foreign Object Debris is considered by the author of this thesis a form of Foreign Object Damage – even when no catastrophic physical damage is imparted a loss in Quality (an economic loss) results from Foreign Object Debris.

1.2.1.3 FOD Taxonomy across Industries

Bird ingestion during flight is publicly viewed as the most widespread instance of FOD; it causes loss of life, economic detriment and concern, albeit only 8% of bird strikes cause damage to aircraft of which 15% are to engines, from which only 32% cause damage to engines [30, 37]. On volume of occurrences, bird ingestion may rank very low in the life cycle of modern aircraft; during production and flight operations FOD in the form of manufacturing byproducts, misplaced tools, environmental debris, personal items, small metallic parts is pervasive [17, 38, 39].

Boeing Company tracks FOD with 9 dedicated non-conformance codes presented in Table 1.1 [38]. FOD data provided by the aircraft division of another large aerospace OEM to the author of this thesis specifies 128 codes used to track FOD. Lockheed Martin Space Systems Company suggest 5 FOD Types/Categories in their FOE Quick-Start Guide as a starting point for their suppliers (see Table 1.2) [40]. The Federal Aviation Administration (FAA) has published a FOD Taxonomy in Advisory Circular No. 150-5210-24 that is presented in Table 1.3 [41]. Beyond aircraft, FOD poses a problem for space and missile systems in those industries – there too taxonomies exist [40, 42].

Table 1.1: Boeing Company Nine FO Non-Conformance Codes

Nine Common FO Non-Conformance Codes	
Degradation	Rust, plastic/seal breakdown etc.
Manufacturing Debris	Metal shaving, loose sealant, rivet tails, etc.
Panstock	Rivets, washers, bolts, screws, pins, etc.
Consumables	Q-tips, caps, bags, tape, rags, cleaners, tie wraps/zip ties, string ties etc.
Personal Items	Pens, keys, change, paper, etc.
Environmental	Dirt, dust, insects, rocks/pebbles, fluids etc.
Tools/ Shop Aids	Wrenches, sockets, screwdrivers etc.
Perishables and Expendables	Clamps, clecos, drill bits, apex tips, etc.
Trash	Plastic wrap, paper backing, used tape, etc.
"FOD Prevention - It Takes a Team", Boeing, NAFPI Conference 2010	

Table 1.2: Lockheed Martin Space Systems Five FOD Types/Categories

Five FOD Types/Categories	
Parts	Any component, assembly, or other item that is installed in the product and intended to be a part of its configuration. Includes excess parts, spare parts, and test parts.
Consumables	Items that are used in the manufacture of the product, but are not intended to be a part of the product's final configuration. Examples of consumable items include tape, zip ties, and cleaning wipes.
Tools	Items that are used to manufacture/test the product such as wrenches, screwdrivers, gauges, mirrors or any piece of one of these items.
Personal Items	Any item that originates from employees that is not normally a part of the production process. Examples include badges, pins, rings, and glasses.
General Debris	True FOD. Any item, particle, or scrap made of any material that represents a potential hazard to the hardware or equipment such as metal shavings, dust/dirt, and strands/fibers of thread or wire.
FOE Quick-Start Guide: "Everything You Need to Build Your Own FOD Prevention Program", Lockheed Martin Space Systems Company 2013	

Table 1.3: FAA FOD Taxonomy

FAA FOD Taxonomy
aircraft and engine fasteners (nuts, bolts, washers, safety wire, etc.);
aircraft parts (fuel caps, landing gear fragments, oil sticks, metal sheets, trapdoors, and tire fragments)
mechanics' tools
catering supplies
flight line items (nails, personnel badges, pens, pencils, luggage tags, soda cans, etc.)
apron items (paper and plastic debris from catering and freight pallets, luggage parts, and debris from ramp equipment)
runway and taxiway materials (concrete and asphalt chunks, rubber joint materials, and paint chips)
construction debris (pieces of wood, stones, fasteners and miscellaneous metal objects)
plastic and/or polyethylene materials
natural materials (plant fragments, wildlife and volcanic ash)
contaminants from winter conditions (snow, ice)
FOD Taxonomy, FAA Advisory Circular No. 150-5210-24

1.2.1.4 Controllable vs. Uncontrollable Sources of FOD

Flight operations (including runway operations) are susceptible to uncontrolled and controllable sources of FOD including, but not limited to: environment, wild life strikes/ingestion, dust, small rocks, sand, aircraft parts/remnants on runways and runway material [17, 30, 36, 37, 41, 43]. The author of this thesis proposes that the distinction between controllable vs. uncontrollable sources of FOD lies in the risk posed by debris to the Primary System during a runway operation (see next section).

1.2.1.4.1 Controllable Source of FOD

A source of FOD is controllable if the risk to the Primary System, the compressor rotor blade in the context of this thesis, is insignificant – explicitly in the context of a runway operation, the risk posed by the debris aspiration/ingestion/impact life cycle from presence on runway to just prior to impact at a rotor blade can still be mitigated, controlled. The risk is mitigated, controlled through countermeasures from the Enabling/Support Systems – prime examples are processes and technologies to minimize debris on the runway that could be aspirated up to the fan face of the engine and to minimize debris

inside the engine that could travel up to the compressor rotor blades and pose a risk of impact.

1.2.1.4.2 Uncontrollable Source of FOD

A source of FOD is uncontrollable if the risk to the Primary System is significant – explicitly in the context of a runway operation, the risk posed by debris that impacts a blade is significant and unavoidable. The risk is mitigated by countermeasures at the Primary System – a prime example being a surface treatment on the rotor blade to improve crack growth resistance.

1.2.2 FOD IS PERVASIVE AND DIVERSE

FOD is pervasive and abundantly diverse during the entire life of modern aircraft; it is also difficult and expensive to deal with – compounding the threat is the reality that 1st level approximations to where and how FOD affects critical, rotating structures such as HPC rotors are completely missing from early design studies of conceptual military jet aircraft – equally needed is the ability of operators of current aircraft to be able to gage the effectiveness and affordability of their operational paradigm and of enhanced configurations. A summary of affected systems and resulting direct/indirect costs will be presented to give a perspective of the holistic impact of FOD.

1.2.2.1 FOD Impact to Commercial Operators

FOD costs the global airline industry $14.86 billion (in 2013 U.S. dollars) PA in direct and indirect costs; $1.35 billion (in 2013 U.S. dollars) PA are direct costs that are burdened with a 10X multiplier from indirect sources such as the forty-nine considered by McCreary [30] (see Table 1.4). Business aside: the world's airlines netted large profits in 2013 of $12.9 billion with total revenues of $708 billion, their Total Operating Costs

(TOCs) are $695.1 with 2.14% of these burdened just by FOD – FOD still outpaced them financially and hardly anyone noticed [31].

The main embodiments of FOD for commercial Operation of modern commercial aircraft are: debris and wildlife strikes to jet engines and tires [30]. Damage to the aircraft skin and/or airframe is not documented thoroughly in the public domain literature, largely because it is economically less costly than damage to tires and/or jet engines – anecdotal evidence suggests that fractionally wing and fuselage damage is 1%-2% of tire and engine damage costs [30]. Ingestion in it of itself may not be harmful, but when the ingested debris or wildlife strikes structure or systems it may cause permanent damage.

Table 1.4: FOD Indirect Costs

Indirect Cost Categories	
IMMEDIATE AIRLINE DELAY COSTS	CREW AND STAFF COSTS
1. Delay for planes in air	24. Lost time and overtime
2. Delay on runway/taxiway	25. Replacement crews required when maximum service hours are reached
3. Delay at gate	26. Reaction by crews leading to disruption of schedule
4. Increased gate or other airport fees	27. Morale and morale-based productivity losses
5. Increased gate staff and ground crew / ground support fees	28. Loss of productivity of injured personnel
6. Cost of aircraft changes	29. Cost of hiring and training replacements
7. Cost of rental or lease of replacement equipment	CORPORATE COSTS
8. Cost of any incident investigation	30. Cost of restoration of orderMedia and PR response
AIR TRAFFIC CONTROL / AIRPORT OPERATIONS COSTS	31. Cost of corrective action
9. Delays to allow for runway/taxiway inspection and cleanup	32. Fines and citations
10. Delays from closed runway	33. Legal fees resulting from passenger action
11. Delays from closed airport	34. Legal fees resulting from airport action
MAINTENANCE AND FLIGHT OPS COSTS	35. Legal fees resulting from action by other airline
12. Fuel efficiency losses	36. Legal fees resulting from action by regulator
13. Unscheduled maintenance	37. Liability claims in excess of insurance
14. Increased work required during ongoing, scheduled maintenance	38. Increased insurance premiums
15. Loss of aircraft	39. Insurance deductibles
16. Increased wear or operating costs on remaining equipment	40. Corporate manslaughter/criminal liability for the corporation
17. Loss of spares or specialized equipment	41. Corporate manslaughter/criminal liability for executives, staff, and pilots/crew
PASSENGER ("PAX") COSTS	REGULATORY COSTS
18. Missed connections	42. Cost of the investigation
19. Rebooking fees and effort	43. Administrative costs of airline reporting, monitoring, etc. to comply with regulator needs
20. Replacement flights on other carriers	ENVIRONMENTAL COSTS
21. Hotels	44. Carbon / Environmental issues
22. Food and meal vouchers	45. Aircraft fuel burned as consequence of delay
23. Loss of business and damage to reputation	46. Ground vehicle fuel burned as consequence of delay
	47. Ongoing aircraft engine fuel efficiency
	AIRPORT RELATED COSTS
	48. Airport efficiency losses
	49. Increased landing fees to pay for resulting airport safety expenditures
Data from "Runway Safety: FOD, Birds, and the Case For Automated Scanning", Insight SRI, 2010	

1.2.2.1.1 FOD to Jet Engines

Wildlife strike damage to jet engines of modern aircraft is overwhelmingly a bird issue – 97% of all wildlife strikes in the U.S. on aircraft from 1990 to 2012 were by birds, of these bird strikes it is estimated from the Federal Aviation Administration's (FAA)

National Wildlife Strike Database Serial Report that 10% caused damage, of that damage 26% was reported on engines [43]. This type of FOD is characterized as Soft Body FOD [17]. McCreary's updated report for cost of FOD to the world's airlines focuses on debris and bird strikes as main sources of Foreign Object Damage – at 2.1 occurrences every 10k movements 8% of bird strikes to commercial airliners cause damage, of which 15% of strikes are to engines, from which 32% of damage events are to engines [30]. The most commonly struck component on a modern commercial airliner is the windshield at 17% of occurrences, but only 6% of these cause damage – the most commonly damaged component are the engines [30]. It is easily calculated using McCreary's data that bird strikes to engines have direct costs of $7,653 per 10k movements (in 2013 U.S. Dollars) – movements are used in the airline industry to refer to each takeoff or landing [30].

Anecdotal evidence hints that the most common mode of failure from wildlife strikes to jet engines is by large plastic deformations, over a large area of component with no material loss, tearing or cracking [17, 30].

Debris-induced damage to commercial airliner jet engines comes in several varieties (see Table 1.3) [30, 33, 36, 41, 44]. The debris that causes damage is typically characterized as Hard Body FOD [17]. The majority of FOD strikes to commercial airliners occur on the runway, at 50% (the remaining 50% is split between the taxiways 30% plus ramps and maintenance areas at 20%) – not at the gates or in the air, yet most airport operators miss 96% to 97% of this type of FOD on the runways because they still rely on visual inspections [30]. Strikes to engines take up 20% of occurrences, with 50% of those causing damage [30]. Size of FOD has direct implications to safety and cost – though less frequent (9% of debris), larger FOD that ranges from 400-500g poses a safety risk as it can

destroy entire engines, primary structure and control surfaces while more frequent (10% of debris), smaller FOD that exists on the runway in piles that range from 2-4g poses a cost risk as it degrades engine efficiency and drives more maintenance/replacement activity [30]. Larger FOD in the form of large pieces of runway material, control surface tracks, fuel caps, hoses and tools poses a grave threat to whole engines and aircraft – but tends to be detected and removed more readily than small FOD [30]. Small FOD in the form of runway material, environmental debris, small parts leftover from other aircraft (including, but not limited to nuts, bolts, washers and other pieces of metal) are ingested by engines as the aircraft move on the runways [30]. McCreary's data can be readily used to calculate a direct cost due to FOD strikes to the engines of $27,452 per 10k movements (in 2013 U.S. Dollars) [30]. Additionally McCreary suggests that as more advanced/complex, lighter, thinner blade designs are used on the turbo machinery of jet engines they will become more prone to small FOD - further driving up replacement and repair activity [30].

Debris strike-induced failure modes on jet engines tend to concentrate on blades – typically the damage is localized and is in the form of material loss, notches and cracks [17, 30, 45]. This damage is largely induced by Hard Body FOD ingestion of small debris – including, but not limited to sand, small rocks, small parts left over from other aircraft, bitumen and parts from the engine that come loose during runway operations [17, 30, 45]. In the context of FOD, on average the jet engine components rejected most during stringent inspections are the compressor and turbine blades [17, 30, 45]. Most of these rejections are due to non-critical cracks on the surface of blade tips, none-the-less the standard practice is that most of these "minor" flaws are "blended in" – this "standard practice" is in of itself an economic failure as it clearly demonstrates how mischaracterized and misunderstood

useful remaining life and risk management at the operator level are. The mechanical detriment does not end there – these FOD-induced small cracks catalyze and exacerbate HCF [17, 45, 46, 47]. In HCF the blade material is weakened by low stress cycles repeated over many cycles, typically over 10^4. FOD induces and worsens HCF through [17, 45, 46, 47]:

- Material erosion on the blades that prompts maintainers to blend-in the surface roughness - now out of balance, the blades' natural frequencies shift to coincide with the engine's natural modes and resonate at stress levels larger than design levels – additionally blending throws off the balance of the engine's turbo machinery leading to fuel efficiency losses of 0.5% to 1.0%
- Strikes on the leading edge of the blades generate residual stress fields that induce faster crack growth - additionally FOD strikes may themselves create very sharp notches that carry very large stress concentrations (K_Ts), these also reduce the life of the blades [17, 46, 47]

1.2.2.1.2 FOD to Tires

The most common items struck by debris in a commercial airliner are the tires – 80% of all debris strikes are to tires [30]. Direct wildlife strikes typically do not damage tires and are not documented in the public domain [30]. The debris striking the tires includes, but is not limited to small parts leftover from other aircraft, rocks and other metallic FOD mostly on runways – 96% to 97% of FOD is missed by the current practice of visual inspections on airport runways [30]. The damage caused by debris strikes to tires has two dominant types: strikes that embed FOD on the tires that causes them to fail retread and strikes that puncture or tear tires leading to their immediate replacement [30]. The most

common forms of visible damage to tires are: cuts or tears (80%), punctures (16%) and gouches (3%) [30]. Tire strikes by debris have an occurrence of 3.2 strikes per 10k aircraft movements (aircraft movements are each takeoff or landing) – more common than bird strikes that have an occurrence of 2.1 strikes per 10k aircraft movements [30]. McCreary's report quantifies the cost of tire strikes at $7,019 (in 2013 U.S. dollars) per 10k aircraft movements [30].

1.2.2.2 FOD Impact to Military Operators

According to USA Today, "Russian warplanes sent to Syria to back the regime of Bashar Assad are breaking down at a rapid rate that appears to be affecting their ability to strike targets, according to a senior Defense official...Nearly one-third of Russian attack planes and half of its transport aircraft are grounded at any time as the harsh, desert conditions take a toll on equipment and crews, said the official who was not authorized to speak publicly about sensitive intelligence matters...The Russians appear to be having difficulty adapting to the dusty conditions, and the number of airstrikes they have conducted seems to have dipped slightly." [154]. It appears that Russian aircraft, long thought of as robust under harsh conditions, are significantly susceptible to FOD (small rocks and sand are considered Hard Body FOD). Costs induced by FOD to military operators are not readily available in the public domain – perhaps because of politics or because of the entwined nature of military acquisitions\expenditures. The militaries' themselves may not be able to understand/quantify/forecast their own overall maintenance costs, let alone those driven by FOD at an aggregate level – interestingly enough, the U.S. Air Force sponsored a report by the RAND Corporation titled "The Maintenance Costs of Aging Aircraft" that focuses on commercial aircraft as a means to glean lessons that may

be applied in the replacement of its own fleets [48]. There is evidence from the literature about how military costs compare with those from commercial operators of modern aircraft – a NATO Research and Technology Organization (RTO) technology report states that costs (inferring from the report as to be direct costs) for military air forces are "comparable" to those from the commercial aviation industry [17]. The NATO technology report states that depending on severity of damage to a jet engine on a military jet by FOD the direct costs to repair are $119,282 - $477,126 vs. McCreary's report presents a direct cost range to repair similar damage to commercial airliner engines of $112,290 - $837,195 (in 2013 U.S. dollars) [17, 30]. A Rough Order of Magnitude (ROM) estimate for the total costs of FOD to military air forces is to take the global commercial airlines' direct costs given by McCreary as $1.35 billion (in 2013 U.S. dollars) PA plus estimated indirect costs attributed to loss of readiness [30]. The $1.35 billion estimate falls between $0.54 billion estimated by Procaccio and Chaplin's estimated $2.47 billion in direct costs burdened by the military air forces of the world due to FOD (in 2013 U.S. dollars) [49, 50].

While there are significant differences between how commercial and military operators employ their aircraft - most notably commercial operators will fly their aircraft for up to ten times as many hours as military operators on a regular basis, but military usage severity is more aggressive and is substantially more detrimental during adverse weather/geographic conditions in the increasingly more common asymmetric combat theaters – there may be a cumulative, smearing effect to damage by FOD.

1.2.2.2.1 FOD to Jet Engines

During aircraft runway operations, early in the takeoff phase FOD damage to engines during runway operations is pervasive and expensive –military operators burden

large economic losses via mitigative, preventive removal and maintenance activities/processes that do not eliminate FOD significantly [17]. On a low bypass turbofan jet engine of a military fighter the literature points to the fan, compressor and turbine blades as the turbomachinery affected most by FOD [17]. The literature evinces that bird and ice strikes, common forms of Soft Body FOD, are most detrimental to forward-most fan blades – overwhelmingly causing large plastic deformations over a wide contact area with little or no material loss or cracking - the literature's focus on fan blades is from a regulatory context, despite evidence that damage from Soft Body FOD such as bird strikes is less frequent and expensive than that from Hard Body FOD [17, 30, 43, 82]. Hard Body FOD is most detrimental to compressor rotor blades and to a lesser extent to turbine blades – the majority of debris found around runway operations is small Hard Body FOD, coupled with a higher rate of impact and damage to engines [17, 30, 93]. The literature evinces that the volume of FOD-induced and/or exacerbated failure modes occur forward-most on low bypass turbofan jet engines – specifically, the HPC rotor blades are susceptible to HCF (phenomena of structural degradation from lower stresses repeated at high cyclic rates), FOD-induced damage and the interaction of the two [17, 46, 80, 81]. A compressor blade specific annual cost estimate can be had from the United States Air Force's HCF Science and Technology Program – a report from the program estimated that the annual cost of HCF per blade to be $110,112 (in 2015 U.S. dollars) (the report estimated the yearly cost of HCF to be $400 million in 2000 U.S. dollars spread over 5,000 blades) – as FOD-induced cracks are primary catalysts and exacerbators of HCF a bulk of this dollar figure is assumed to be attributable to FOD impact at compressor blades [17, 45, 46, 47, 127, 128]. Aside from posing a serious operational/structural risk to engines and to aircraft

readiness, the foot print of economic detriment due to FOD-induced damage is large enough to cover the acquisition and operational costs of several weapon systems – eliminating the FOD-induced failure mode presents a sizeable opportunity to improve affordability and operational readiness concurrently.

1.3 IMPROVED AFFORDABILITY VIA FOD ELIMINATION

Although not the most commonly struck component, impact of compressor rotor blades by Hard Body FOD exacerbates HCF, the largest driver of maintenance activity and associated costs on jet engines of military aircraft – sizeable real estate in the effort to drive improved affordability [17, 46, 80, 81, 83, 84]. This thesis will be aligned specifically to the elimination of the FOD-exacerbated HCF failure mode caused by Hard Body FOD impact damage on blades on the 1st rotor disk of the HPC of low bypass, turbofan engines of military aircraft during runway operations early in the takeoff phase – where damage in a LCC context ranges from Line Replaceable Unit (LRU) repairs to replacements and associated economic detriment - in the context of this thesis the LRU and the Primary System are the compressor rotor blade. FOD Elimination is hereby defined as eradication and robustness toward controllable and uncontrollable sources of FOD respectively in the Enabling and Primary Systems - robustness in this context is the ability to function without failure under various conditions while eradication is avoiding/ridding sources of FOD.

1.3.1 WHAT'S NEEDED TO ELIMINATE FOD?

Current efforts (FOD Programs for prevention and removal of debris on the runway) of operators of modern aircraft, focused on preventing and removing FOD, do not systematically consider the root causes that lead to it, nor do they consider Primary System robustness; the bulk of the efforts are reactive in nature (inspections, "Clean as You Go",

detection, removal) – these efforts are not looped back to early design studies of conceptual aircraft [33, 41, 44]. A theoretical framework is presented in Figure 1-5 that synthesizes a novel a methodology and process to achieve affordable, FOD Elimination on current and future engine/engine on aircraft designs. The new approach's key enablers are:

1. Quantifying and characterizing FOD during runway operations and associating those findings under a taxonomy that identifies dominant drivers.
2. Leveraging Industrial Engineering and Economics methods and theories to develop analytics to gage economic effectiveness of elimination methodologies
3. Leveraging system synthesis approaches from Systems Engineering to develop processes where FOD Elimination methodologies operate/iterate
4. Leveraging physics-based formulations and approaches from Fracture Mechanics, Probabilistic Fracture Mechanics, Robust Engineering to develop methodologies to gage required robustness to FOD
5. A literature review on the industry's paradigm for "managing" FOD reveals:
    - Characterization/Quantification of FOD from Operators, military and commercial – to find trends, commonalities and best practices
    - Paradigm's technologies, processes and design inefficiencies
    - Adjacent and remote disciplines that offer potential applications and theories to enhance/complement FOD Elimination
    - Laws, standards and market analysis

The intent is that from this framework a methodology and a process will arise that will assess how affordably portfolios of technologies, processes and design improvements

eliminate FOD. From the theoretical framework presented in Figure 1-5 it is evident that a contribution to the state of the art of Systems Engineering for design of modern aircraft will be had by the infusion and/or revised application of concepts from Industrial Engineering and Economics – concepts such as Conformance Levels will be adapted to operate on FOD Elimination goals by trading-off technologies, processes and design improvements with associated LCCs. An additional benefit apparent in the theoretical framework will be the infusion of Fracture Mechanics into the methodology for FOD Elimination – an infusion of a discipline entirely missing in the early stages of design.

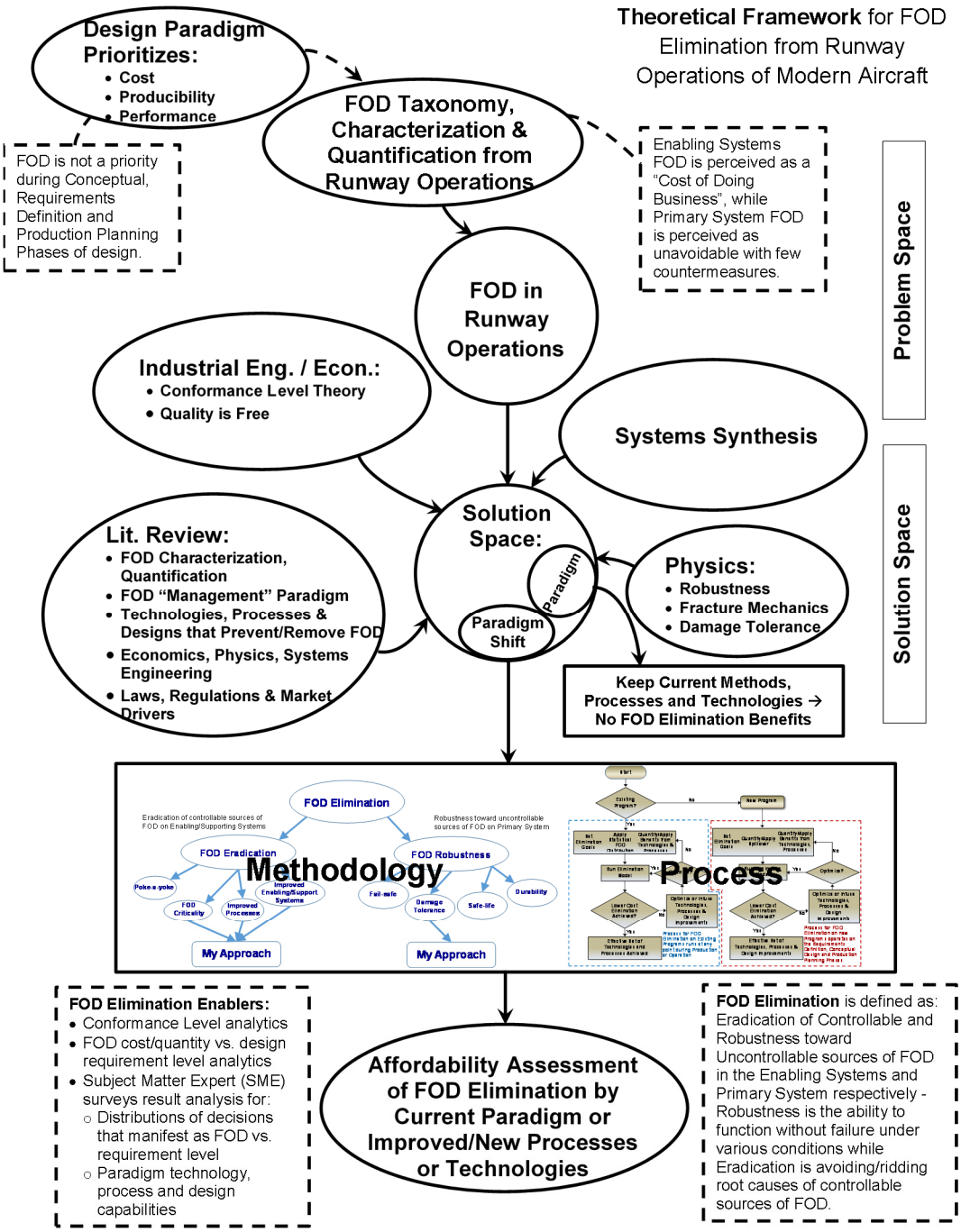


Figure 1.5: FOD Elimination Theoretical Framework

1.3.2 Research Objective

From the Theoretical Framework a research objective is synthetized: to determine if FOD elimination, the minimization of the FOD-exacerbated HCF failure mode that leads to 1st HPC rotor blade replacements and/or catastrophic engine losses while reducing or

marginally increasing LCCs, is achievable and if so by what mix of technologies, processes and/or design changes at the primary and support systems.

2 BACKGROUND/MOTIVATION

2.1 CURRENT FOD PREVENTION/REMOVAL PROGRAMS

FOD Prevention and Removal programs for commercial and military aviation are mandated – both operators and OEMs abide by the same or similarly derived guidelines and standards focused on operation, OEMs additionally have to abide by Quality standards for aerospace [41, 44, 55, 56, 57, 58]. The detailed standards and guidelines are distilled into actionable guidelines or programs by organizations such as the National Aerospace FOD Prevention, Inc. (NAFPI) and NATO's Research and Technology Organization (RTO) – these programs may then be tailored by OEMs and Operators to improve their own or to adapt to their production or operations requirements for FOD prevention and removal [17, 33]. An additional sobering reason for their existence in commercial aviation is a market that is accustomed to increasingly safer flight (see Figure 2-1) – in the context of FOD this ideal of ever safer flight may be reaching a paradox since commercial aircraft will double in numbers by 2032 while bird populations are growing, airport growth will be slower and new flight destinations may be in locations more prone to FOD [30, 59, 60] (see Figure 2-2 and Figure 2-3). On the military aviation side FOD prevention and removal programs will continue to exist and grow because current and future combat operations will be more diverse in terms of geography/topology, asymmetry and aircraft usage in more FOD prone areas (see Figure 2-4) – increasingly posing a greater risk of FOD to aircraft is a more aggressive and frequent combat usage of these fewer, consolidated military assets [17, 33, 61, 62]. Over time stakeholders of FOD prevention and removal programs accrue large amounts of valuable data on locations, densities and sizes of debris on a runway – that tribal knowledge is not currently mined for consideration during early design studies

of conceptual aircraft – currently, in the public domain a methodology leverage that valuable data is missing.

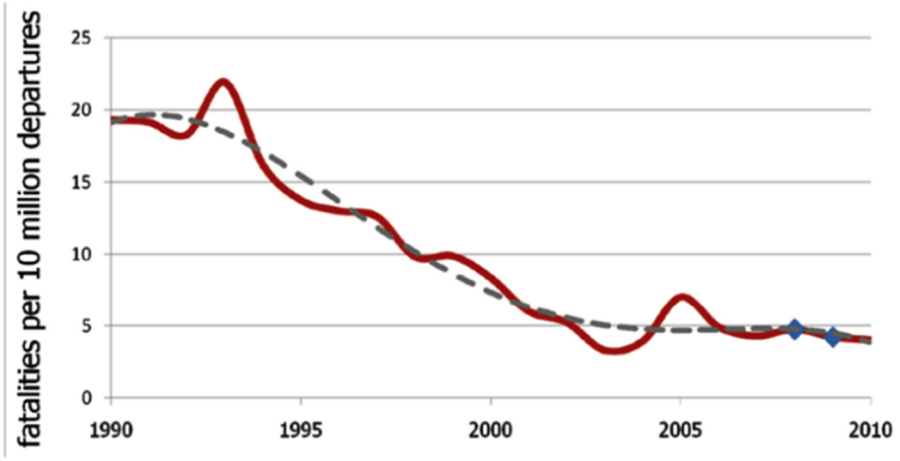


FIGURE A: ICAO data for the worldwide number of accidents involving fatalities for 1990 – 2010* among aircraft weighing more than 2250kg (MTOM>2250) *(data, RED; trend line GREY DASH, 2009 and 2010 estimated).*

Figure 2.1: Fatalities per 10 m Departures

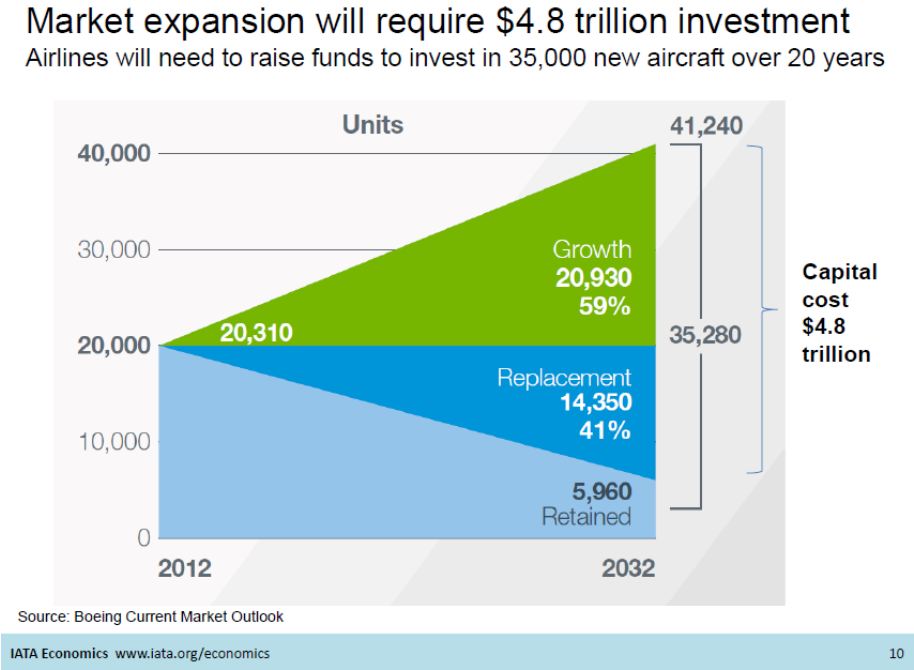


Figure 2.2: Global Commercial Fleet Doubling by 2030

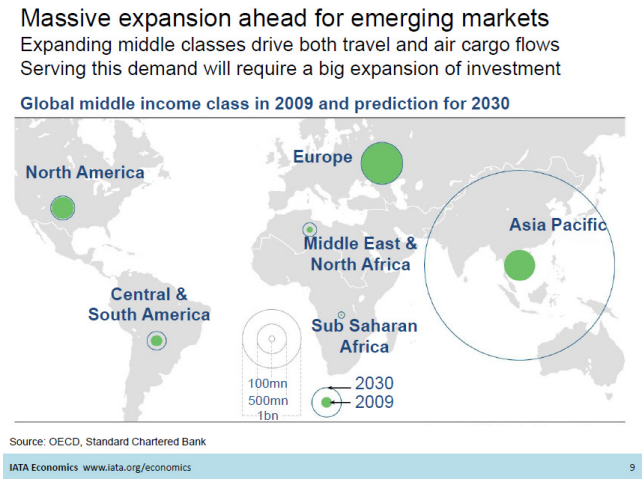


Figure 2-3 Commercial Aviation Expansion into Emerging Markets

Figure 2.3: Evolving Geography/Topology of War

2.2 FOD Elimination Implications from Runway Operations

OEMS and operators will be able to leverage a systematic/symbiotic methodology and process to address the expensive and possibly catastrophic consequences of damage to the HPC rotor blades due to impacts from Hard Body FOD on the runway early in the takeoff phase by:

- Infusing and/or revising the application of concepts from other disciplines – concepts such as Conformance Levels will be adapted to operate on FOD Elimination goals by trading-off technologies, processes and design improvements vs. LCCs [51, 63, 64] – a side benefit will be the ability to develop knowledge early on in the design process related to operational FOD
- Infusing Fracture Mechanics – leveraging disciplines such as Probabilistic Fracture Mechanics and Damage Tolerance to characterize and quantify structural damage caused by debris impact on rotor blades and to assess the benefits of Primary System countermeasures such surface treatments to retard/halt structural degradation

The risk from Hard Body FOD to aircraft during runway operations is felt economically – great economic opportunity is at hand by its elimination. FOD Elimination has a market potential of at least $16.35 billion ($14.86 billion direct + direct costs from commercial aviation and $1.35 billion calculated direct costs of military aviation in 2013 U.S. dollars PA) PA in savings to military and commercial operators globally [17, 30].

Safer travel and combat would be a significant output from FOD Elimination. As commercial and military aviation are projected to grow in the next two decades operating aircraft under a new framework that determines how controllable sources of FOD can be mitigated by Enabling/Support Systems and how robust the Primary System needs to be toward uncontrollable sources of FOD will reduce engine susceptibility to critical failures and reduce LRU replacements and repairs.

FOD Elimination would have the consequence of improving aircraft brand since perceived Quality is a beneficial component of perceived value – literature and data in this

thesis makes a compelling business case that OEMs and operators of aircraft are very sensitive to FOD, it is very likely they will perceive aircraft that are rid of controllable and are robust to uncontrollable FOD as high Quality, high value products [65].

2.3 HOW IS FOD DEALT WITH TODAY?

Currently, Original Equipment Manufacturers (OEMs) and operators of military aircraft with low-bypass turbofan engines do not consider the expensive and possibly catastrophic consequences of damage to the HPC rotor blades due to impacts from FOD during runway operations early in the takeoff phase in a systematic/symbiotic practice – even 1st level approximations to the affect/effect of FOD to these critical, rotating structures are completely missing from early design studies of conceptual military jet aircraft – equally important is the need by operators of current aircraft to be able to gage the effectiveness and affordability of their operational paradigm and of enhanced configurations. The presence, distribution and nature of FOD on the runway are analyzed separately from its structural or economic degradation to turbomachinery of the engine. Missing in the public domain is a symbiosis of the physics and probabilistic models for how FOD on the runway is aspirated, how and where it is then ingested by the engine, what trajectory through the turbomachinery does it follow to impact the rotor blades, where and at what velocities do impacts occur and ultimately what is the ensuing damage and its effect on crack growth at the location of the impact sites. Also missing in the public domain is a synergistic assessment of the LCC impact of the current methodologies for mitigating the deleterious effects from FOD on the runway – equally needed is a methodology to gage the LCC implications of improved or new processes, technologies and design

improvements to mitigate the effects of FOD on current and future engine/engine on aircraft designs.

FOD is "managed" in a case-by-case fashion – as stated in a previous section commercial and military operators take guidelines from organizations like NAFPI and suit their perceived needs in terms of a FOD Program [33]. Most FOD Programs follow a pattern of operation where prevention and removal are the dominant activities – detection becomes a precursor to removal and evaluation is used to improve the programs post-facto [33, 36, 38, 40, 42, 44, 55, 66, 67, 68] (see Figure 2-5). The spillover value of decades of knowledge and experience on FOD "management" from dozens of aircraft is not looped back to early design studies – the importance of design is briefly highlighted in the National Aerospace Standard 412 (NAS412) for FOD Prevention and the Product Realization section of the Aerospace Standard AS9100C for Quality Management Systems in Aviation/Space/Defense organizations [56, 69]. AS9100C states, "Where appropriate, the organization shall divide the design and development effort into distinct activities and, for each activity, define the tasks, necessary resources, responsibilities, design content, input and output data and planning constraints" – this directive conflicts with systematically looping back to the early design phases in an integrated manner – integrating FOD Elimination into the design activities will drive eradication of controllable sources and will impart robustness to uncontrollable sources of FOD. Technical aside: if the holistic impact of FOD on the Primary System is assessed to be significant during early design of new engines on aircraft or on current designs, the author of this thesis proposes that **FOD Criticality** be adapted as a call-out in design drawings or operational specifications for components prone to FOD. FOD Criticality is hereby defined as the attention/due diligence

that must be placed on the design and/or operation of a system, subsystem or process whose significant degradation or failure due to FOD would pose a severe economic, mission or safety risk to the Primary System.

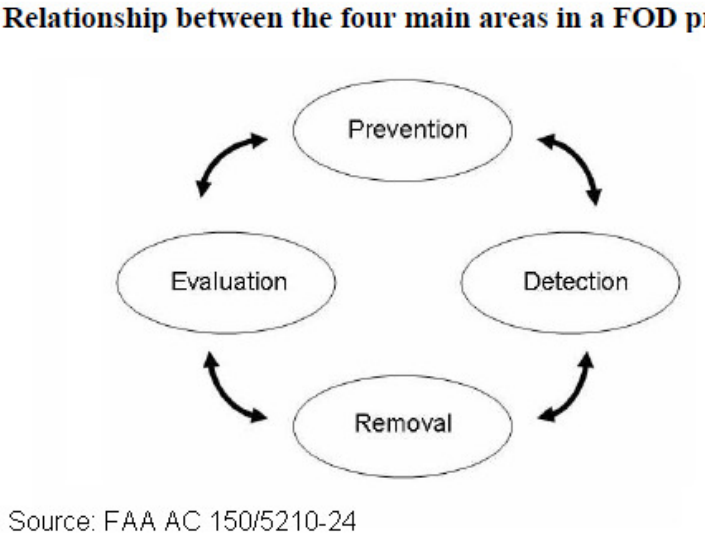


Figure 2.4: Four Main Areas in a FOD Program

2.3.1 FOD PREVENTION, DETECTION, REMOVAL & EVALUATION

NAS412 is widely used today by commercial and military OEMs and operators as a guideline or model for FOD Programs; NAS412 provides a rigorous/thorough set of activities put in place to prevent FOD that is representative of the current paradigm, its key points are highlighted in the ensuing subsections [56]. Commercial and military operators

2.3.1.1 FOD Program Administration

At the top of a FOD Program for OEMs and operators executive commitment from the organization's leadership sets the tone for impact and. Organizational leadership also identifies the FOD Program leader that takes the strategic direction they set and converts it to actionable measures and policy – key to the role of the program leader is the authority, resources and ability to set the tone for a "FOD averse" culture for the entire organization.

As needed a FOD Prevention Committee may be constituted cross functionally – the committee supports the program leader in promoting the culture of FOD prevention, reviewing/assessing the program, pinpointing areas of program improvement/growth, supporting the development of specialized techniques/methods of prevention, supporting FOD investigations, characterizing/quantifying types and sources of FOD and assisting in developing training material. The FOD Area Focal Point (Focal) is responsible for validating the effectiveness of the FOD Program, actively participating in the FOD Prevention Committee, serving as POC for the program and associated training and recommends the level of FOD awareness that must be assigned to an area. Managers/Supervisors are responsible for FOD Program adherence to policy, FOD prevention and Tool Control procedures, requirements for FOD or lost tool investigation, root cause analysis, area visit control, making management and/or the FOD program leader aware of prevention needs, keeping training current and reviewing/assessing suitable metrics for compliance and/or trend analysis.

2.3.1.2 Area Designations

FOD Areas are designated and/or categorized in the FOD Program in order to control the susceptibility of equipment/hardware/processes prone to FOD in varying levels - the most severe area of vigilance being the shadow of the aircraft. As the risk by FOD increases prevention activities become more aggressive and rigorous – area criticality levels assigned according to risk convey level of awareness, participation and detail that must be placed to prevent and remove FOD from the equipment/hardware/processes – NAS412 lists the following categories of area criticality level in ascending order of risk

potential and attention, “General Housekeeping, FOD Awareness, FOD Control and FOD Critical”. Runways are FOD Critical areas.

2.3.1.3 Debris/FOD Migration

In the context of this thesis only risk from FOD migration in the form of Hard Body FOD left on the runway from other aircraft is considered. Debris may pose a risk during runway operations if it is of significant size and quantity on a runway. If FOD migration from the areas of generation to the vicinity of equipment/hardware/processes susceptible to it is not controlled the risks associated with it increase. The methods to control FOD migration are extensive making it impractical to try to list/encompass them; these methods may include but are not limited to: encapsulated/arrested small miscellaneous parts or fasteners, caps, protective films, barriers, receptacles, consumables control, fool-proofing “Poka-yoke”, etc.

2.3.1.4 Clean-As-You-Go and Housekeeping

Clean-As-You-Go is a legacy concept that has stood the test of time – quite simply it calls for clean-up during and after work is performed. Housekeeping is a means to establish effective processes to prevent the migration and/or cross-contamination of FOD – it ensures that areas are kept free of debris.

2.3.1.5 Consumables, Expendables & Hardware

NAS412 states that in the context of Management of Consumables, Expendables and Hardware “The purpose of this process is to establish defined practices for the accountability, storage, use/consumption, and disposal of consumables, expendables, and hardware for effective control within the product areas to prevent the risk of these items being left within the product as FOD” [56]. The techniques employed to prevent

consumables, expendables or hardware from becoming FOD include, but are not limited to maintaining a clean and organized workstation, proper labeling and identification of consumables and expendables, removal of spent or expired consumables or expendables such as adhesives or fastener stems, isolating and/or grouping parts from assemblies when they are disassembled, Point of Use storage to control quantities of consumables, expendables or hardware to only what is required, use of spill or loss proof containers, item or tool control methods and technologies, enhanced tool or item accountability methods and technologies, control of administrative items that could become FOD (staples, rings, pencils, etc.) and special consideration should be given to the use of "poka-yoke", the Japanese term used to refer to "mistake-proofing", to make areas/workstations unable to produce or migrate FOD (see Table 2.1) [71].

Table 2.1: NAS412 Example of FOD Risk and Control Method

Potential Risk	Possible Control Method
Rivet washer	Magnetic nose pieces
Rivet stem	Containment bags
Trimmings from safety wire/cable, cotter key	Pliers and dykes rubber inserts
Debris from drilling, sanding, grinding, polishing	Vacuum collection systems

Source: NAS412

2.3.1.6 Tool Management

In the context of this thesis tools do not pose a significant risk to compressor rotor blades as they are very likely to be detected and removed from a runway and in the case they were not, they very likely will not be ingested across the fan face – they would impact the fan and/or stators/guide vanes before they would impact an HPC blade. As stated by NAS412, "The essence of Tool Management is a process that controls and accounts for all tools used in manufacturing, test, assembly, and operational environments to prohibit tools from causing damage to the product". Activities that prevent tools from becoming FOD

include but are not limited to: limiting/controlling tools to the adequate quantity for the task(s) at a specific work area (controlling what was checked out, by whom and for what purpose – accountability through ownership/responsibility), use of tool containers, control of personal items, procedures to replace/remove tools that are out of specifications or become damaged, lost tool procedures to conduct search, investigation, documentation and close-out, tool storage or kitting, tool storage and/or location indicators, tool marking and/or chitting, advanced tool design for specific uses, RFID tags to track/monitor tool checkout and location, automatic inventory and marking for magnetic resonance to track/find lost tools.

2.3.1.7 Material Handling and Packaging

In the context of this thesis material handling and packaging do not pose a significant risk to compressor rotor blades. NAS412 states that for effective Material Handling and Packaging, "FOD prevention controls built into material handling and storage operations will help prevent the introduction of contamination". Specific processes have to be put in place properly store, handle and move material that is required in production and flight line activity for aircraft; such processes include, but are not limited to employee training for compliance with handling/storage/shipping requirements, material handling/parts protection procedures, accountability and reporting processes and kitted protection devices.

2.3.1.8 Product and Process Design Considerations

Product and Process Design Considerations are activities and/or processes that are put in place to avoid migration, collection or entrapment of FOD where risk is heightened. Some of these activities and/or processes include avoiding/reducing the volume and

quantities of cavities or enclosures where FOD could be trapped, minimization of FOD migration prone geometries, designing-in access panels for inspections and/or removal of debris, captive hardware, use of fasteners that are design to not generate FOD, seal-out moisture or debris, design inlets or ducts to avoid environmental FOD, design production or operation processes to avoid FOD generation, employ devices to avoid, divert or filter-out FOD and design tools and equipment to avoid generating, migrating or entrapping FOD.

2.3.1.9 Aerodromes

The section on Aerodromes in NAS412 is extensive and the reader is directed to read it to get a thorough understanding of what it takes to prepare, monitor, assess and maintain the environment around operating aircraft relatively free of FOD. As stated in NAS412, "Foreign objects can be generated from personnel, airport infrastructure (pavements, lights, and signs), the environment (wildlife, snow, ice) and the equipment operating on the airfield (aircraft, airport operations vehicles, maintenance equipment, fueling trucks, other aircraft servicing equipment, and construction equipment)" – sources for FOD are diverse and complex. Keeping the aerodrome safe to operating aircraft requires a concerted set of activities that range from understanding the local types and sources of FOD, the immediate weather, aircraft types, tools, environmental contaminants, runway types, material and condition, contractor activities, ingress and egress procedures, aircraft servicing, airfield maintenance, foreign object detection at the aerodrome and use of inspections and sweeps.

2.3.1.10 Bird/Wildlife Strikes

In the context of this thesis bird/wildlife strikes do not pose a significant risk to compressor rotor blades – the literature surveyed did not provide an instance where damage to rotor blades from ingested birds or wildlife occurred – damage to compressor rotor blades from external debris will be shown in a subsequent section to be the purview of Hard Body FOD. As stated by NAS412, "The intent is to safeguard air operations assets and flight crews by preventing bird and wildlife strikes with aircraft" – as was previously discussed in this document, avoiding collisions/strikes with birds and wildlife is paramount in keeping aircraft operationally safe and affordable. Activities and processes put in place to avoid bird/wildlife strikes include but are not limited to: wildlife hazard assessments, management of the environment that attracts wildlife to the aerodrome, exclusion techniques to keep wildlife out of aerodrome, active monitoring of bird/wildlife, understanding migrations of animals, designing-in flexibility to flight schedules to minimize risk of strike, active harassing/repelling or birds or wildlife, radar monitoring/tracking of animals around aerodrome, aircraft design and strike reporting for trend analysis.

2.3.2 FOD Management for Runway Operations

The FOD Program at an airport will include a specific protocol for management of FOD on the runways. The typical protocol in the United States for runways employed by commercial operators is to inspect visually 1 to 2 times per operational 24 hour day – the Federal Aviation Administration mandates one inspection a day but most large airports require 2 inspections per operational 24 hour day [30]. The inspection of the runway is done visually on a vehicle by two personnel driving at 20 mph [30]. This inspection process is able to find and remove only 3% - 4% of FOD on the runway per operational 24 hour

day [30]. Runways at military bases in the United States are inspected 1 time per operational 24 hour day in the same manner as their commercial counterparts, visually on a moving vehicle with similar inspection/removal performance – Moffett Field in California goes through a single, visual inspection every morning [70]. In the context of this thesis the assumption will be made that the current paradigm for FOD management at an airport requires 2 daily visual inspections performed by two personnel on a moving vehicle with the ability to detect and remove 3% - 4% of runway debris per operational 24 hour day. The literature also supports the assumption that commercial and military runways are swept with mobile or towed debris removing equipment 1 time per operational 24 hour day – mobile or towed debris removing equipment is mandated by the FAA to remove a minimum of 90% of FOD in the path of the sweep at a minimum speed of 15 mph [30, 41].

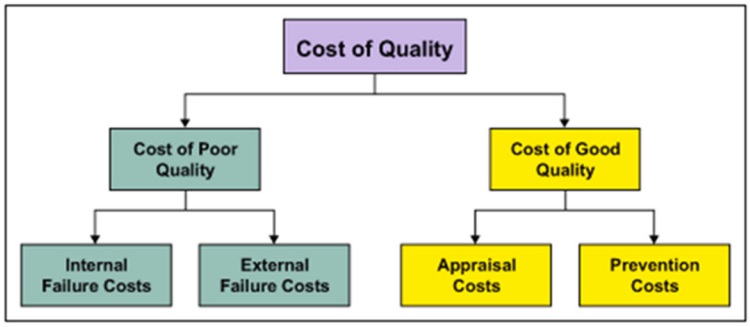
2.4 CURRENT FOD PREVENTION/REMOVAL PARADIGM DEFICIENCIES

Specifically for runway operations early in the takeoff phase, the current paradigm for prevention and removal of FOD does not systematically/symbiotically consider the expensive and possibly catastrophic consequences of damage to the HPC rotor blades due to impacts from Hard Body FOD. The presence, distribution and nature of FOD on the runway are analyzed separately from its structural or economic degradation to turbomachinery of the engine – the discipline that characterizes and quantifies the consequences of Hard Body FOD impact at the rotor blades, Fracture Mechanics, is disconnected from the economy of runway operations. The current paradigm is reactive - no public domain evidence could be found to corroborate that tradeoffs between current and improved technologies, processes and design improvements vs. LCCs for holistically mitigating the effects of FOD had been conducted. The current paradigm of prevention and

removal relies heavily on manned visual inspections for FOD on runways – despite very low performance in terms of detection and removal [30]. The personnel that can significantly influence the consequences of FOD, those that inspect and sweep runways are disconnected from the economics of debris impact to turbomachinery because metrics that link their effort to FOD mitigation performance are not available.

The current paradigm for FOD prevention/removal does not make a direct case to pursue robustness to uncontrollable sources of FOD – sources of FOD that pose an impact risk that cannot be eliminated entirely and thus must be design for or impart robustness toward them at the Primary System [56]. The current paradigm is a balance between the cost of "good quality" and the cost of "poor quality" (referred to as Prevention & Appraisal and Internal & External Failure Costs respectively in the literature) – non-conformance, allowing a significant amount of debris on the runway, is willingly accepted along with the tradeoff between costs of prevention vs. detection as a means to "economically" address runway FOD [72, 73, 74]. The breakdown of Cost of Quality presented in Figure 2-5, broken-out into the activities that contribute to cost of "poor quality" and "good quality" in Table 2.2, aides a better understanding of Quality in the context of FOD during runway operations for military, low-bypass, turbofan jet engines. The aerospace industry, OEMs and operators, is an oligopsony and an oligopoly, few buyers and few sellers – the literature hints that this oligopsony-oligopoly co-dependence supports the aerospace industry's unwillingness to shift away from the Economic Conformance Level (ECL) paradigm, that ever careful balance of cost of "good quality" vs. "poor quality" – in the context of FOD Elimination the industry spends a great deal of money and effort on manual/reactive

processes and not enough on automated/proactive improved processes and new technologies [72, 74, 75, 76, 77, 78, 79] (see Figure 2-6).



Source: http://www.isixsigma.com/implementation/financial-analysis/cost-quality-not-only-failure-costs/

Figure 2.5: Cost of Quality

Table 2.2: Cost of Quality Breakdown

Flight Operation Delays	Repairs due to FOD	Planning for FOD Prevention	Checking and Testing that Runway Operations are on Track for FOD Elimination
Primary System Redesign / Runway Operations Changes	Performance/Economic Degradation	Engineering Design Review	Runway FOD Audits
Runway Inspections (manual & automated)	Loss of Vehicle	FOD Prevention Improvement Meetings	Technology or Process Audits
Unscheduled Corrective Action Activities	Loss of Life	FOD Prevention Improvement projects	Tool and Item Control / User Accountability Process for Runway
Reporting Lost Tool/Items on Runway	Unable to Support the War Fighter Mission	FOD Prevention Education and Training	Clean as You Go
		General Housekeeping Practices	
Internal Failure Costs are costs that are caused by products or services not conforming to requirements or customer/user needs and are found before delivery of products and services to external customers. They would have otherwise led to the customer not being satisfied. Deficiencies are caused both by errors in products and inefficiencies in processes. **External failure Costs** are costs that are caused by deficiencies found after delivery of products and services to external customers, which lead to customer dissatisfaction. **Prevention Costs** are costs of all activities that are designed to prevent poor quality from arising in products or services. **Appraisal Costs** are costs that occur because of the need to control products and services to ensure a high quality level in all stages, conformance to quality standards and performance requirements. Source of Definitions: http://www.isixsigma.com/implementation/financial-analysis/cost-quality-not-only-failure-costs/			
Modified by author in the context of FOD Elimination at Compressor Rotor Blades During Runway Operations for Military, Low-Bypass, Turbofan Jet Engines.			
Originally provided as source of Quality Costs in Context of FOD at OEMs by Mark J. Rodriguez, FOD SME.			

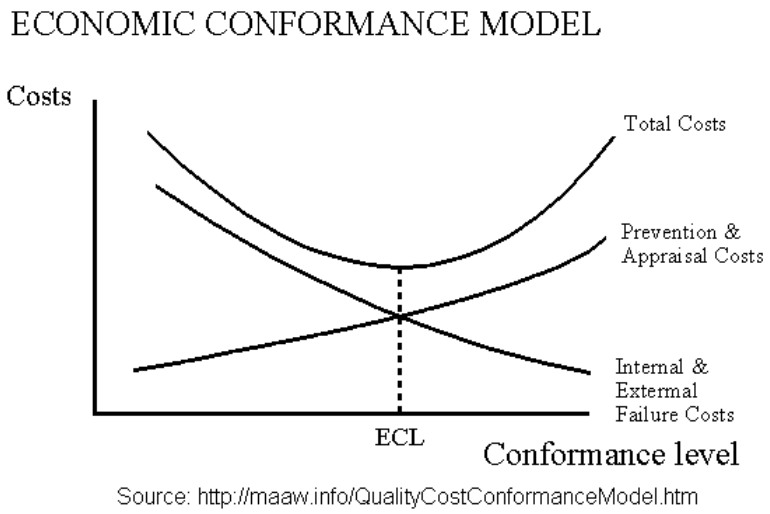
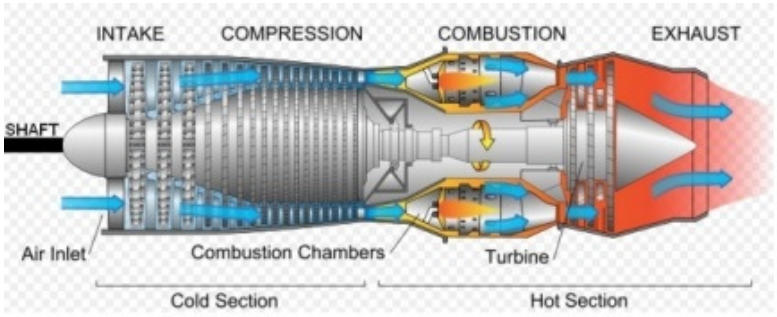


Figure 2.6: Economic Conformance Level Model

2.5 WHERE *IT* HURTS MOST – FOD DETRIMENT ON AN ENGINE MAP

As was presented on previous subsections of this thesis, FOD damage to engines during runway operations is pervasive and expensive – commercial and military operators burden a great deal of economic losses via mitigative, preventive removal and maintenance activities/processes that do not eliminate FOD significantly [17, 30]. So where on an engine map is FOD most impactful? The literature points to the fan, compressor and turbine blades being affected by FOD on modern jet engines [17, 30]. The literature evinces that bird and ice strikes, common forms of Soft Body FOD, are most detrimental to forward-most fan blades – overwhelmingly causing large plastic deformations over a wide contact area with little or no material loss or cracking [17, 30, 43]. Hard Body FOD is most detrimental to compressor rotor blades and to a lesser extent to turbine blades – the majority of debris found around runway operations is Hard Body FOD, coupled with a higher rate of impact and damage to engines [17, 30]. On an engine map several failure modes occur at each section –Figure 2-7 presents a low bypass turbofan engine and along with Table 2.3 a mapping of the most common failure modes arises [17, 46, 80, 81]. Fan blades, though

forward-most on an engine, are mostly susceptible to Soft Body FOD – the literature's focus on fan blades is from a regulatory context, despite evidence that damage from Soft Body FOD such as bird strikes is less frequent and expensive than that from Hard Body FOD [17, 30, 82].



Pratt & Whitney F135-PW-600 F-35B STOVL Engine
www.airforce-technology.com/features/featurelockheed-martin-f35-pratt-whitney-engine-analysis/

Figure 2.7: F135-PW-600 Engine

Table 2.3: Engine Failure Modes

	Failure Modes / Component	1	2	3	4	5	6	7	8
Compressor	Rotors and Stators	HCF	FOD	HCF+FOD Interaction	Erosion	Corrosion	Clang	Fretting	
	Disc	Fatigue	Creep	Wear	Rubbing				
	Compressor Tie Bolts	Mechanical Fatigue	Wear	Fretting	Rubbing				
Combustor	Liner	Mechanical Fatigue	Fretting	Buckling	Wear	Thermal Fatigue	Yield Slip	Thermal Distortion	Corrosion
	Casing	Fatigue	Pressure Cycles						
	Cross Fire Tubes	Wear	Rubbing	Fretting	Corrosion	Thermal Fatigue			
	Transition Piece	Thermal Fatigue	Wear	Rubbing	Fretting				
Turbine	Rotors	HCF	Creep	Corrosion	Sulphidation	Erosion			
	Stators	Creep Rupture	Corrosion	Sulphidation	Bowing	Fatigue	Thermal Fatigue		
	Rotor Disc	Creep Rupture	LCF						

2.6 CASE FOR RESEARCH ON FOD ELIMINATION

2.6.1 FOD Elimination Definition & Research Alignment

In the context of this thesis, FOD Elimination is defined and aligned respectively as follows:

- Eradication and robustness toward controllable and uncontrollable sources of FOD respectively in the Enabling and Primary Systems - robustness in this context is the ability to function without failure under various conditions while Eradication is avoiding/ridding controllable sources of FOD
- This research effort is specifically aligned to the elimination of damage caused by Hard Body FOD on blades on the 1st rotor disk of the HPC of low bypass, turbofan engines of military aircraft during runway operations early in the takeoff phase – where damage in a LCC context ranges from LRU repairs to replacements and associated economic detriment - in the context of this thesis the LRU and the Primary System are the compressor rotor blade

2.6.2 FOD Elimination - Low Bypass, Turbofan Rotor Blades

The literature and the mapping presented on Figure 2-7 and Table 2.3 demonstrates that the volume of FOD-induced and/or exacerbated failure modes occur forward on engines – specifically, the HPC rotor blades are susceptible to HCF, FOD-induced damage and the interaction of the two [17, 46, 80, 81]. The emergent case for research on the elimination of Hard Body FOD from low bypass, turbofan engine HPC rotor blades of military aircraft during runway operations is further evinced by the following [17, 46, 80, 81, 83, 84]:

- Low Cycle Fatigue (LCF), phenomena where blade material subjected to high stress cycles at frequencies generally lower than 10^4 cycles, is less frequently a cause of failure on modern jet engines as compared to HCF (see Figure 2-8)
- HCF, a phenomena where blade material is weakened by low stress cycles repeated many times (typically begins in the range of 10^5 to 10^7 cycles), is the prime driver of component failure in modern jet engines of military aircraft and is a significant concern for future engines – FOD is a key catalyst of HCF failures as follows (see Figure 2-9 and Figure 2-10):
    - Strikes on the leading edge of the blades generate residual, tensile stress fields that induce faster crack growth
    - FOD strikes themselves create sharp notches that carry large stress concentrations (KTs), these also reduce the life of the blades
    - Material erosion by small Hard Body FOD drives operators to blend-in surface roughness - now out of balance, the blades' natural frequencies coincide with the engine's natural modes and resonate at stress levels larger than designed for
    - HCF/FOD interaction is evident: less time to reach unstable flaw sizes and high cycle, low stresses are amplified further reducing component life – reductions of orders of magnitude vs. undamaged blades (see Figure 2-11)
- Additionally the blending by operators to mitigate material erosion due to small Hard Body FOD throws off the careful balance of the jet engine's turbo machinery leading to fuel efficiency losses 0.5% to 1.0%

- Exacerbating the effect of FOD on HCF are technological limits to current inspection capabilities – notches caused by FOD on blades are often smaller than can be inspected for – as blades accumulate many cycles small flaws initiate and grow cracks to failure (see Figure 2-8)
- Compressor section provides 55 – 60 % of total engine thrust – any efficiency and/or structural decay degrades overall engine performance
- A paradigm shift is needed to steer OEMs and operators toward elimination – current approach to mitigate FOD is not holistic, nor is FOD field data driving design activities – the current standards and guidelines for OEMs and operators are reactive and disconnected from a Systems Engineering practice

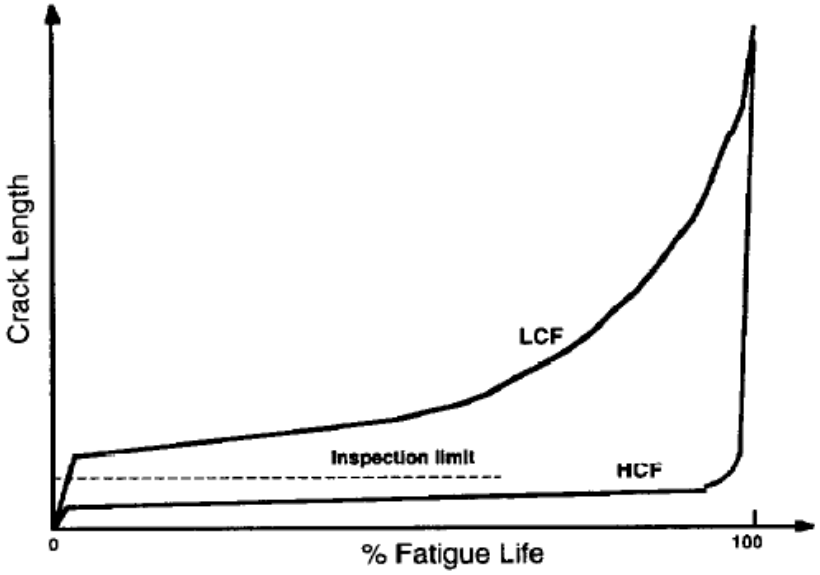


Fig. 1 Schematic showing difference between HCF and LCF.

"High Cycle Fatigue Management in Gas Turbine Engines", NATO Research and Technology Organization (RTO), MP-17, October 1998

Figure 2.8: HCF vs. LCF

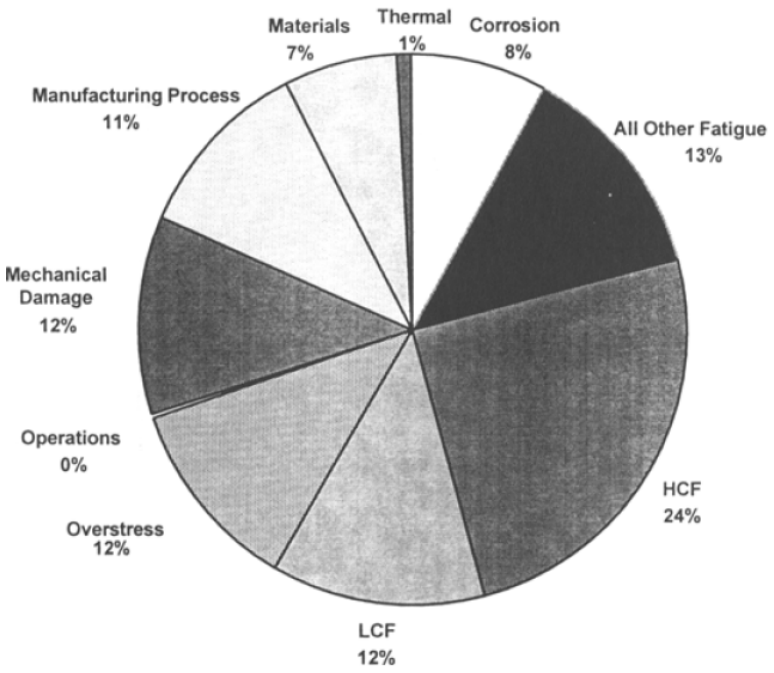


Figure 1. Representative jet engine component distress mode statistics.

Cowles, B.A., "High cycle fatigue in aircraft gas turbines - an industry perspective", International Journal of Fracture, Volume 80, Issue 2-3, pp 147-163, 1996.

Figure 2.9: Military Aircraft Jet Engine Failure Modes

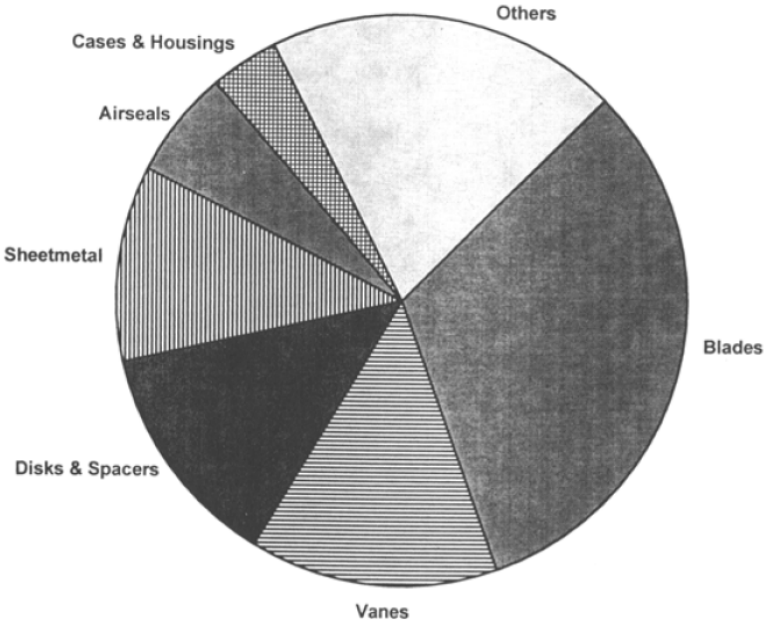


Figure 2. Representative distribution of HCF problems by component.

Cowles, B.A., "High cycle fatigue in aircraft gas turbines - an industry perspective", International Journal of Fracture, Volume 80, Issue 2-3, pp 147-163, 1996.

Figure 2.10: Component Failure Due to HCF Distribution

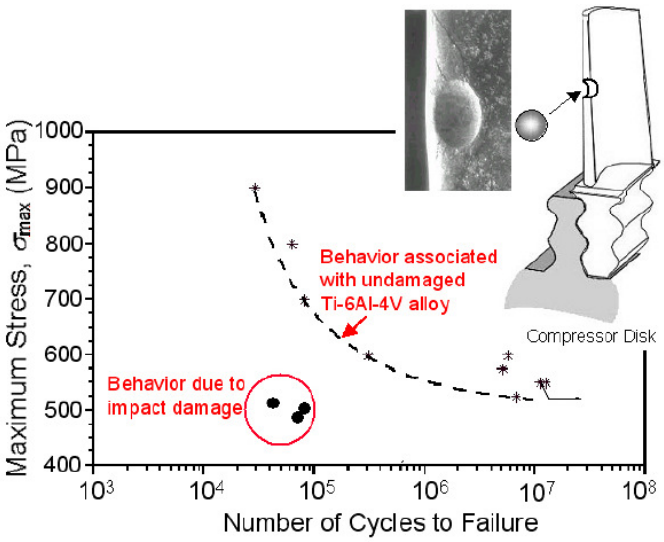


Fig. 1. Stress-life fatigue behavior of undamaged Ti-6Al-4V compared to impact damaged material (3.2 mm dia. spherical hardened steel projectile fired at 200-300 m/s). Load ratio, R = $\sigma_{min}/\sigma_{max}$ =0.1.

"Synchrotron Mesodiffraction: A Tool for Understanding Turbine Foreign Object Damage", Stanford Synchrotron Radiation Laboratory, 2003

Figure 2.11: Fatigue Life Reduction Due to FOD

2.7 ACHIEVING FOD ELIMINATION

A case on the elimination of damage from Hard Body FOD from low bypass, turbofan engine HPC rotor blades of military aircraft during runway operations in the early takeoff phase has been made – a methodology and process is now required to achieve elimination. Elimination will drive structural life improvements to HPC rotor blades by eliminating damage from FOD and its deleterious effect on HCF (see Figure 2-12).

	Failure Modes / Component	1	2	3	4	5	
Compressor	Rotors and Stators	HCF	FOD	HCF+FOD Interaction	Erosion	Corrosion	Clan
	Disc	Fatigue	Creep	Wear	Rubbing		
	Compressor Tie Bolts	Mechanical Fatigue	Wear	Fretting	Rubbing		

Figure 2.12: Elimination Benefits on Component Failure Mode

The case for eliminating FOD was made clear in previous sections, the consequences of not achieving the goal are dire indeed:

- Continued operation of aircraft/engines under a paradigm that is inefficient and expensive
- Continued design of engines/engines on aircraft that are not FOD robust and will operate increasingly in FOD prone areas around the globe – recall that Russian aircraft in the Syrian theater of operation had a 40% drop in readiness due to FOD from desert conditions (refer to Section 1.2.2.2)
- The consequences of FOD are not the cost of doing business as was said by a top executive at an OEM – a 1st level approximation to the affect/effect of FOD can be made by the right mix of physics/probabilistics and Systems Engineeing – better designs may be the outcome of implementing the methodology set forth in this thesis
- Large LLC savings could be missed by not pursuing FOD elimination systematically as it is showcased in this thesis – the cost of not being able to perform missions, of increased spares and unscheduled blade replacements and of perceived unrobust engine/engie on aircraft designs are likely to be large

2.7.1 Morphology of Elimination

A progression chain for FOD present on runway damage on a compressor rotor blade is presented on Figure 2-13:

Figure 2.13: Foreign Object Debris to Damage Progression

The progression chain of Foreign Object Debris to Damage lends itself to conceptualizing the morphology for achieving FOD Elimination – the form is presented in matrix fashion in Table 2.4, while the function will be the stuff of ensuing subsections. The numerical/analytical experimental permutations to be considered are circled in Table 2-4.

Table 2.4: FOD Elimination Morphology

FOD Elimination Progression Chain			Elimination Alternatives / FOD Progression Criteria	1	2	3	4	5	6
	Eradication of FOD by Enhancing Enabling/Supporting Systems	Prevent/Remove	Prevent/Remove Runway FOD	Improve Runway Sweep	Runway Visual Inspections, Current & Improved	Runway Automatic Scanning	Debris Barriers	Improve Management of FOD from other A/C	
		Avoid	Avoid FOD Ingestion	Improve Engine Inlet Location	Improve Engine Inlet Geometry				
		Avert	Avert FOD Impact	Filtering Devices	Diverting Devices	Engine Duct Shape	Sacrificial Structure		
	Robustness at the Primary System Toward Damage from Remaining FOD After Eradication Enhancements	Eliminate	Eliminate FOD Damage	Geometric Changes to Blade	Blade Surface Coatings	Blade Surface Treatments	Blade Material Changes	Blade Performance Changes	Damage Tolerance Interval Engine Inspection

2.8 IMPLEMENTING FOD ELIMINATION

2.8.1 MODEL FOR FOD ELIMINATION

With the context for FOD Elimination presented previously and with a focused research direction/alignment a model for elimination is presented in Figure 2-14. The

model has the goal of enabling FOD Elimination by preventing/removing the presence of debris, avoiding its ingestion by the engine and eliminating its damage by providing robustness toward debris that does impact blades.

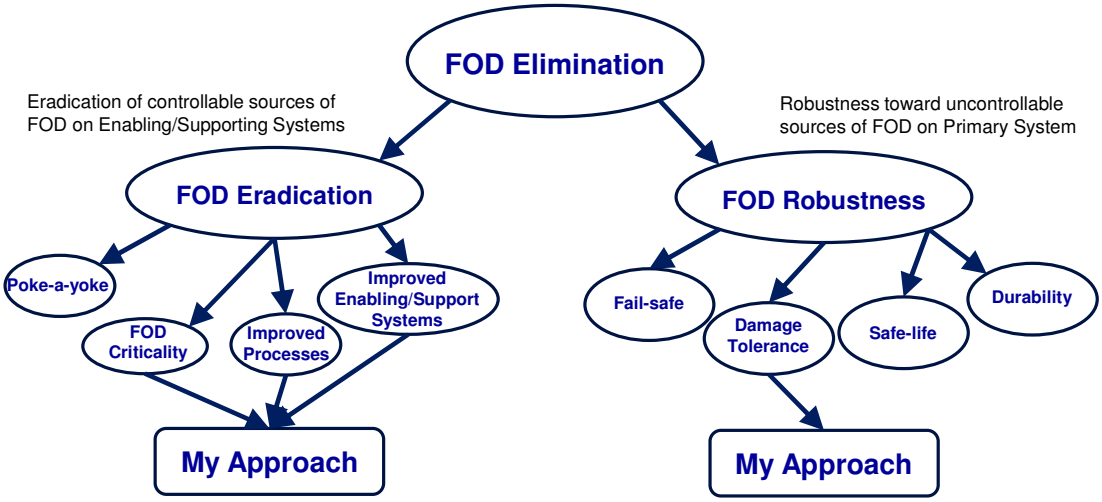


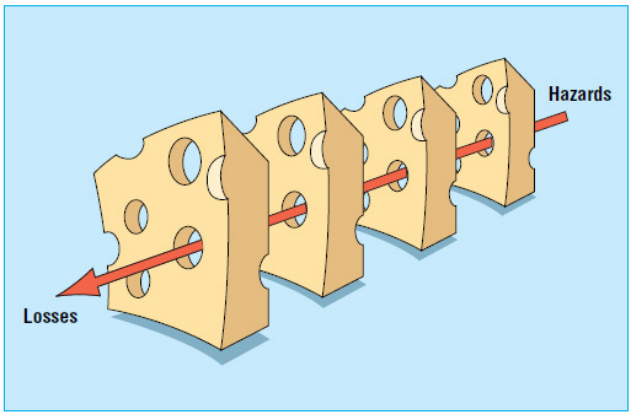
Figure 2.14: FOD Elimination Model

2.8.2 Methodology for FOD Elimination

The methodology for FOD elimination presented previously in this thesis will include modifications to accommodate the focused direction/alignment of eliminating FOD-induced damage on HPC rotor blades. The proposed methodology stems from Taguchi's concept that products should be designed to be on target and robust to uncontrollable sources of noise and from Orlandella and Reason's Swiss Cheese Model that propounds accident progression across layers of defense in a system (see Figure 2-15)[85, 186]. The resulting methodology is as follows:

- In the FOD debris to damage progression chain enhancements must be made to prevent/remove the presence of debris, avoid ingestion and mitigate the effects of impact on the blades

- The assumption is made that layers of defense or eradication (prevent/remove and avoid ingestion) will have weaknesses, thusly allowing some debris to impact and damage the blades – robustness to debris that makes it across the defense layers must be designed into the blades to mitigate the effects of impact – in this context design modifications, such as surface treatments, specifically align to robustness



The Swiss cheese model of how defences, barriers, and safeguards may be penetrated by an accident trajectory

"Human error: models and management", BMJ Volume 320, pp 768-770, 2000.

Figure 2.15: Swiss Cheese Model

2.8.2.1 Formulae for Gaging FOD Elimination

The methodology to gage FOD elimination aggregates a Measure of Elimination (M_oE) from a Measure of Eradication (M_oER) and a Measure of Robustness (M_oR).

2.8.2.1.1 Measure of Eradication (M_oER)

A Measure of Eradication (M_oER) aggregates non-dimensionally the efficacy of processes and/or technologies at ridding FOD on the runway. Embedded in the formula is the proposed process by which M_oER may be achieved, e.g., the current paradigm to reduce debris on the runway requires: one daily sweep of the runway that takes close to an hour and has an efficacy of collecting 90% of debris over the area and at the time of sweep, two daily visual inspections by vehicle that each take 30 minutes and have an efficacy to

detect/collect 80% of debris during the time of inspection. The equation for M_oER for one FOD type non-dimensionally accounts for remaining debris on the runway after processes and/or technologies are applied during one operational 24 hour day (see Equation 1). To more readily understand the model for M_oER Figure 2-16 presents the numerator of Equation 2 as a uniform distribution of FOD through-out the operational day - McCreary's report on the cost of FOD to commercial airlines drives the assumption that FOD through-out the day is uniformly distributed and very quickly returns to an equilibrium as debris accumulates and distributes [30]. The number 1 in the formula represents the 100% of debris that is present instantly at any time in the operational day. Processes and/or technologies for the length of time and area they are engaged, how often they are employed and associated efficacies reduce the amount of runway debris left on the runway.

$$M_oER_{FOD_i} = \left[\frac{24 \cdot 1 - \sum_{DRPT}(N_{DRPT} \cdot T_{DRPT} \cdot \eta_{DRPT}) - \sum_{IPT}(N_{IPT} \cdot T_{IPT} \cdot \eta_{IPT,Removal} \cdot \eta_{IPT})}{24 \cdot 1}\right]_{FOD_i} \quad (1)$$

Where:

i – specifies debris type

DRPT – debris removal process and/or technology

N_{DRPT} - how many times a debris removal process and/or technology is engaged

T_{DRPT} – duration of debris removal process and/or technology procedure

η_{DRPT} – efficacy of removal process and/or technology

IPT – inspection process and/or technology

N_{IPT} - how many times an inspection process and/or technology is performed

T_{IPT} – duration of inspection process and/or technology procedure

η_{IPT} – detection efficacy of inspection process and/or technology

$\eta_{IPT,Removal}$ – efficacy of removal during inspection process and/or technology

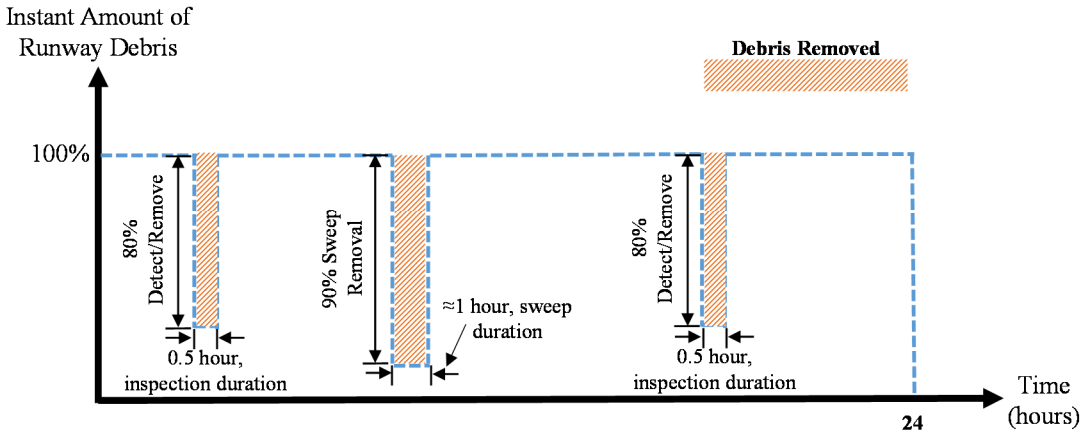


Figure 2.16: Debris on Runway for Operational Day Model

2.8.2.1.2 Measure of Robustness (M_oR)

A Measure of Robustness (M_oR) (see Equation 2) is proposed based on the number of Primary System (rotor blade) replacements. The intent of robustness is for the Primary System to operate under various conditions, i.e. uncontrollable sources of FOD, the assumption is made that a robust Primary System would operate fully up to its design life. This implies that Primary System replacements are a means to measure overall robustness – specific to the case study to be considered later in this thesis, the replacement reductions that could arise from design improvements and/or technology infusions such as Laser Shock Peening (LSP) at the blade surface and Damage Tolerance interval engine inspections (vs. the current operator paradigm of no engine inspections until the first depot level teardown engine inspection).

The modeling required to calculate the number of Primary System replacements is complex (including deterministic and probabilistic formulae) – as such, the M_oR becomes contextual, it depends on the number of replacements (resulting from a numerical/analytical experimental setting) that the baseline replacements will normalize.

The baseline encompasses the current paradigm process, design and/or technologies experimental factor settings for Primary System replacements.

$$M_o R_{FOD_i} = \left[1 - \frac{\sum_{RMT} N_{replacements,RMT,l}}{N_{replacements,baseline}}\right]_{FOD_i} \quad (2)$$

Where:

i – specifies debris type

RMT – robustness method and/or technology

l – DOE setting

$N_{replacements,RMT,l}$ – Primary System replacements with a robustness method and/or technology at a DOE setting

$N_{replacements,baseline}$ – Primary System replacements from baseline configuration

Since the Primary System, the compressor blades, may encounter more than one type of FOD an aggregate, normalized Measure of Elimination must be considered. Additional consideration must also be given to the value an operator may place on eradication or robustness enhancements – this is done via expert-weighted aggregation of Measures of Eradication and Measures of Robustness. The equation for Measure of Elimination for more than one FOD type is proposed as follows (see Equation 3):

$$M_o E = w_{ER} \cdot \left\{1 - \frac{\sum_{i=1}^{p} M_o ER_{FOD_i}}{p}\right\} + w_R \cdot \left\{1 - \frac{\sum_{i=1}^{p} M_o R_{FOD_i}}{p}\right\} \quad (3)$$

Where:

w_{ER} – weight of eradication

w_R – weight of robustness

2.8.2.2 Formulae for Probability of Achieving FOD Elimination

As there is a debris to damage progression chain, there is also a probability of damage chain that is described by Equation 4 for a FOD type and in Equation 5 for multiple types. Public domain commercial operator evidence presents field data on how often debris is found on runways, frequency of impacts and how often it causes damage [30]. Military operator evidence in the public domain presents a limited set of data for frequency, location and magnitude of damage [17]. Missing in the public domain is a mapping, in the radial and chord-wise direction, where impacts on the blades occur – most impact tests on blade representative coupons focus on the leading edge, a misconception that in a later section will be debunked [87, 88, 89, 90]. Also missing in the public domain are the impact velocity distributions and a mapping of these to where they impact rotor blades – the public domain lists velocity ranges for impact, but no link to where these impacts occur nor what magnitude of impact velocity occurs at those contact sites [17, 87, 88, 89, 90, 91]. In an aggregate context, direly missing is how velocities, locations and ensuing impact flaw geometry and size are connected – some relationships exist in the public domain for impact velocity and depth of impact penetration, but not how they relate to locations on rotor blades [90, 91, 92]. The location, incident velocities and geometry/size of impact flaws (depth and width of penetration) on the blade are key in the modeling of crack growth as a means to estimate the probability of significant damage – damage that would warrant a Primary System replacement. Later sections in this thesis will present deterministic and probabilistic models to estimate debris particle aspiration from runway to engine fan face, in-engine kinetics, impact velocities and locations, impact flaw depth and width and ensuing crack growth at impact sites. Equation 6 presents the end goal – an estimate of the probability of achieving FOD elimination.

$$P_{Damage_{FOD_i}} = P_{FOD_i Present} \cdot P_{Aspirated/FOD_i Present} \cdot P_{Ingested/Aspirated} \cdot$$

$$P_{Impact/Ingested} \cdot P_{Damage/Impact} \quad (4)$$

$$P_{Damage} = \frac{\sum_{i=1}^{p} P_{Damage_{FOD_i}}}{p} \quad (5)$$

$$P_{FOD\ Elimination} = 1 - P_{Damage}$$

where:

FOD_i – specific Foreign Object Debris type

$P_{FODiPresent}$ – probability that Foreign Object Debris is present on runway

$P_{Aspirated/FODiPresent}$ – probability that if a Foreign Object Debris particle is present on runway it will be aspirated to the engine fan face – that it will travel from the ground to engine fance via ground vortex action

$P_{Ingested/Aspirated}$ – probability that if a Foreign Object Debris particle is aspirated to the engine fan face it will travel through 1st stage low pressure compressor fan and reach the 1st stage HPC rotor disk (particle travels through fan without impact)

$P_{Impact/Ingested}$ – probability that if a Foreign Object Debris particle is ingested it will impact a rotor blade on the 1st stage HPC rotor disk

$P_{Damage/Impact}$ – probability that if a Foreign Object Debris particle impacts a blade on the 1st stage HPC rotor disk it will cause significant damage – damage that would warrant a replacement of the blade

It will become apparent in later sections of this thesis that some probabilities will be affected by the processes, technologies and design improvements used to reduce the ammount of debris on the runway and eliminate damage to the rotors.

2.8.2.3 Cost Modeling

In a later section of this thesis an aggregate model will be presented that estimates LCCs from runway inspections and sweeping, repairs and replacements of rotors, aircraft fuel efficiency degradation and associated acquisition, labor and operation for a representative engine subjected to a ferry flight spectrum.

2.8.2.4 Overarching Modeling and Simulation Architecture

The overarching architecture that will enable modeling and simulation of portfolios of technologies, processes and design enhancements that may affordably drive FOD Elimination is presented in Figure 2-17. Embedded in this architecture is the process where the methodology for FOD Elimination operates. This process may be employed assess the current paradigm or to optimize or enhance existing or new compressor rotor blade designs respectively for FOD Elimination.

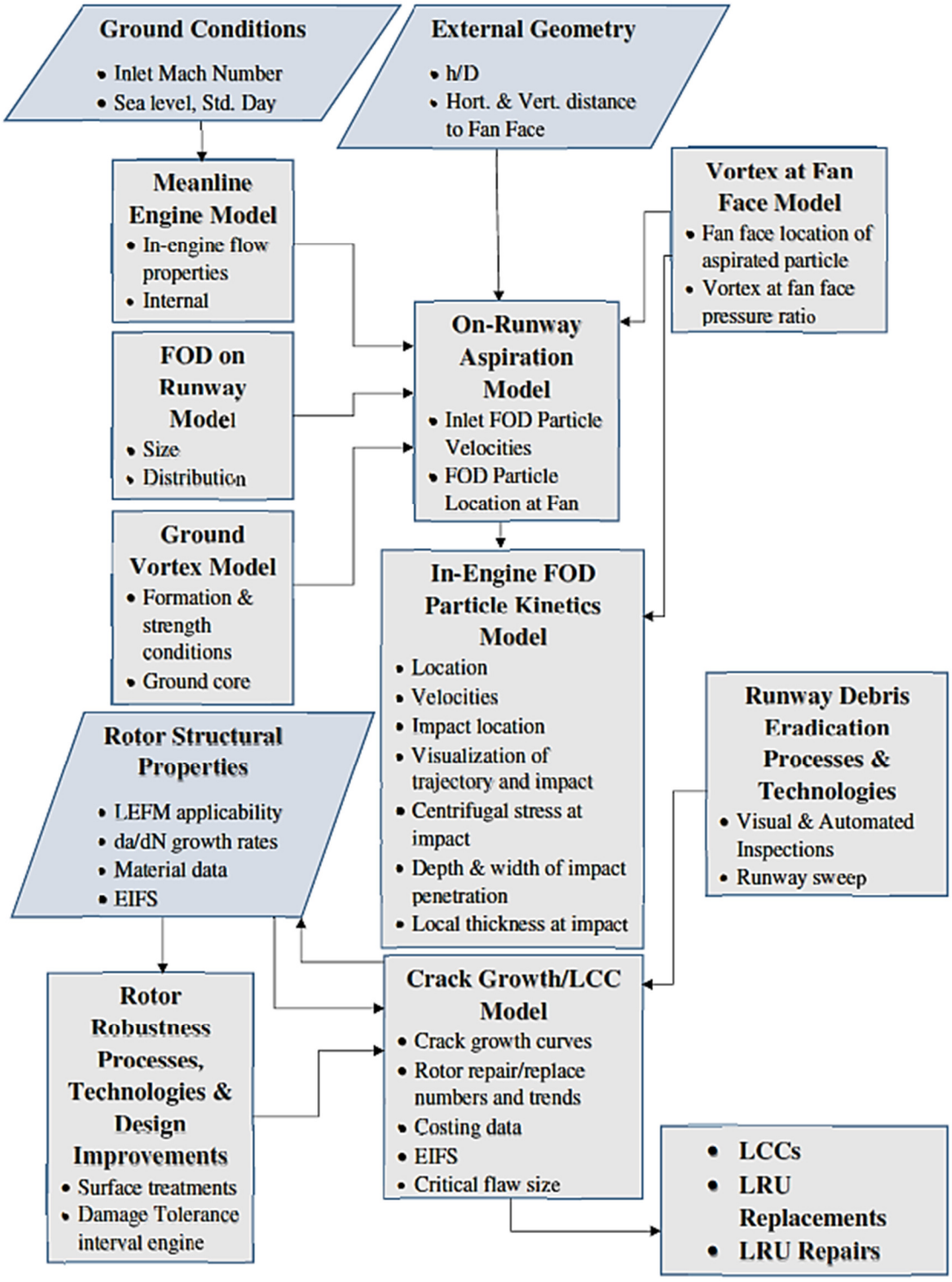


Figure 2.17: Modeling and Simulation Architecture

3 RESEARCH QUESTIONS, HYPOTHESES & SUCCESS CRITERIA

The previous sections of this document built-up a case for the need of research on FOD Elimination – a case that presents the great volume of resources spent on mitigating, not eliminating this expensive and dangerous nuisance following a traditional Economic Conformance Level (ECL) model – an economic model that in its current form may be in conflict with the market's serious focus on lifecycle affordability. Recall the two definitions of affordability from the International Council on Systems Engineering (INCOSE) and the National Defense Industry Association (NDIA):

- INCOSE definition – "Affordability is the balance of system performance, cost and schedule constraints over the system life while satisfying mission needs in concert with strategic investment and organizational needs" [2].
- NDIA definition – "Affordability is the practice of ensuring program success through the balancing of system performance (KPPs), total ownership cost, and schedule constraints while satisfying mission needs in concert with long-range investment, and force structure plans of the DOD" [3]. KPPs are Key Performance Parameters.

Affordability in the context of this thesis becomes the balance of cost and system performance – specifically, the balance of LCCs to eliminate the FOD-exacerbated HCF failure mode and to reduce LRU (the blade) replacements. Elimination of FOD, in the context of this thesis, has the primary goal of minimizing the physical and economic impacts of this phenomena – its definition is recalled:

- Eradication of controllable sources of FOD in the Enabling System – recall that a source of FOD is controllable if the risk to the Primary System is insignificant –

explicitly in the context of a runway operation, the risk posed by the debris from presence on runway to just prior to impact at a rotor blade can still be mitigated, controlled

- Robustness toward uncontrollable sources of FOD at the Primary System – recall that a source of FOD is uncontrollable if the risk to the Primary System is significant – explicitly in the context of a runway operation, the risk posed by debris that impacts a blade is significant and unavoidable

In the context of affordability for this thesis FOD elimination becomes minimizing the debris impact - exacerbated HCF failure mode that leads to Primary System replacements and/or catastrophic engine losses while reducing or marginally increasing LCCs. This overall context of eradication of controllable and robustness toward uncontrollable sources of FOD and affordability leads the way to the ensuing observations, research questions, hypotheses and success criteria:

Observation 1:

For compressor rotor blades FOD is complex in terms of diversity of debris from Supporting Systems (e.g. the runway) and ensuing imparted failure modes and economic impact. The OEM and operator paradigm is to mitigate, not eliminate, the deleterious, costly effects of FOD – it is commonly accepted as "the cost of doing business". The paradigm, via current standards and guidelines separates, not integrates, efforts to mitigate FOD – ensuring that the problem will not go away because there is an entwined nature between the inefficiencies of current processes and technologies that prevent and remove debris and designs that are not robust to debris that eventually impacts a blade and causes damage. A concerted Systems Engineering effort is needed to:

- Prevent/remove debris on the runway and avoid impact on HPC blades
- Eliminate damage ensuing from debris that does impact HPC blades

Based on this observation the research question is as follows:

Research Question 1:

Can elimination of the FOD-exacerbated HCF failure mode be achieved affordably in the life cycle of an engine?

Hypothesis 1:

The elimination of the FOD-exacerbated HCF failure mode can be achieved affordably in the life cycle of an engine. A methodology was constructed to estimate the effectiveness of portfolios of technologies, processes and design changes at affordably eliminating damage exacerbated by Hard Body FOD on HPC rotor blades of low bypass, turbofan engines of military aircraft during runway operations early in the takeoff phase. The methodology will calculate LCCs and rate of LRU replacements associated with the:

- Aggregate efficacy of technologies and processes that eradicate Hard Body FOD on runways – eradication via technologies and processes that prevent/remove debris and avoid its ingestion
- Aggregate efficacy of technologies, processes and design improvements that provide robustness toward Hard Body FOD impact at the HPC rotor blades

By employing the methodology to perform tradeoffs of processes, technologies and design changes that affect the efficacy of eradication and robustness vs. LCCs and rate of LRU replacement the most affordable portfolio that eliminates the FOD-exacerbated HCF failure mode will be identified/contrasted vs. the cost and rate of LRU replacements of the

current paradigm of a single sweep and two visual inspections of the runway once per day to remove debris with no design changes to rotor blades and no interval engine inspections.

Success Criteria 1:

Success is achieved if the methodology estimated LCCs are marginally higher or lower and LRU replacement rates are lower vs. the cost and rate of LRU replacement of the current paradigm of a single sweep and two visual inspections of the runway once per day to remove debris with no design changes to rotor blades and no interval engine inspections.

Observation 2:

A decision maker often is hampered from doing all that can be done because she has budget constraints and does not have enough knowledge ahead of time to make a decision. If a decision maker had a means to determine if investing in a new technology vs. optimizing the current paradigm would eliminate FOD damage to engines more affordably she would be able to more strategically allocate resources.

Research Question 2:

By employing the proposed methodology can the countermeasures that need to be deployed at the Support System and what/where countermeasures at the Primary System be identified that achieve FOD elimination affordably?

Hypothesis 2:

Yes by leveraging the proposed methodology the countermeasures that need to be deployed at the Support System and what/where countermeasures at the Primary System will be identified that achieve FOD elimination affordably.

Success Criteria 2:

Success is achieved if by employing the methodology what mix and placement of countermeasures estimates LCCs that are marginally higher or lower and LRU replacement rates that are lower vs. LCCs and rate of LRU replacement of the current paradigm.

Observation 3:

The public domain literature presents evidence that the magnitude of damage to engine components scales with the size of debris on the runway, though no scientific relationship is made. If in addition to impactor size other experimental factors influence the affordability of FOD elimination their significance must be gaged.

Research Question 3:

Is there a subset of experimental factors that influence LCC affordability more significantly due to FOD elimination in comparison to other factors that were also varied during the numerical/analytical experiments?

Hypothesis 3:

Yes there are experimental factors that influence LCC affordability more significantly due to FOD elimination vs. other factors that were also varied during the numerical/analytical experiments. A set of experiments will be run to gage the effectivity of the methodology at affordably eliminating FOD – a Design of Experiments will establish the manner in which the experimental factors (engine operation, engine external/internal geometry, particle size, runway debris sweeping, modes of inspections and countermeasures) will be varied. A screening test will be conducted to reveal the importance of the factors on the variability of impact occurrence, blade location, impact velocity, LCCs, and LRU replacements/repairs.

Success Criteria 3:

Success is achieved if the screening test reveals statistical significance (P-value test vs. null hypothesis) for factor setting and output (impact occurrence, blade location, impact velocity, LCCs, and LRU replacements/repairs).

4 MODELING & SIMULATION

4.1 MORPHOLOGY OF DAMAGE, ELIMINATION & MODELING

The Oxford dictionary defines morphology as the study of the form of things. Section 2.6 demonstrated that FOD is a key catalyst of HCF – where HCF is the prime driver of component failure in modern jet engines of military aircraft and is a significant concern for future engines. With a focus on FOD catalyzed HCF a morphology for damage arises and is presented in Table 4.1. Encircled in the morphology are the alternatives that address each phenomena represented by the criteria. The morphology for damage is followed by a morphology for elimination in Table 4.2 (also presented in section 2.7.1). In this morphology alternatives are presented to address FOD at stages of eradication and robustness (see Section 1.3). The ensuing subsections will present the background, assumptions and physics that enable the elimination of FOD.

Table 4.1: Morphology of FOD-Induced Damage

Damage Progression Chain <----------	Alternatives / Criteria	1	2	3	4	5	6	7	8	9	10	11
Eradication of FOD by Enhancements	Runway Debris	Metallic	Bitumen	Rocks	Spherical	Irregular Shape	Flat	Uniform Distribution/ Density	Grouped	<2g	1.33 mm diam	3.2 mm diam
	Arrival at Fan Face	Aspiration by Ground Vortex Action	Lofting	Structural Mishap								
	Ingestion	FOD particle travels through LPC fan										
	In-engine Trajectory to HPC Rotor	Particle kinetics - no collision with turbomachinery ahead of HPC rotor	Ricochet at turbomachinery ahead of HPC rotor									
Damage Tolerance & Robustness at Primary System Toward Damage from Remaining FOD After Eradication Enhancements	HPC Rotor Impact	Airfoil side	Chord-wise location	Radial location	Plastic Deformation	Material Loss/Crack	Thumbnail Crack	Part Through Crack	Notch	Erosion		

Table 4.2: Morphology of FOD Elimination

FOD Elimination Progression Chain			Criteria \ Alternatives	1	2	3	4	5	6	7
Damage Tolerance & Robustness at Primary System Toward Damage from Remaining FOD After Eradication Enhancements	Eradication of FOD by Enhancements	On Runway	Prevent/Remove Runway FOD	Sweeping Process	Visual Inspections	Automatic Scanning	FOD Walk	Runway Integrity	Aircraft Integrity	Ramp to Runway Rolling Takeoff
		In Engine	Avoid FOD Ingestion	Improve Engine Inlet Location	Improve Engine Inlet Geometry					
			Avert FOD Impact	Filtering Devices	Diverting Devices	Engine Duct Shape	Sacrificial Structure			
		Control/Sustain	Control/Sustain FOD Damage	Structural Redesign	Surface Coatings	Surface Treatments	Advanced Materials	Damage Tolerance Engine Interval Inspections		

Table 4.3: Morphology of Models

Model Progression Chain	Criteria \ Alternatives	1	2	3	4	5
	Debris Aspiration to Fan Face via Ground Vortex Action	Benoulli Mechanics for Ground to Fan Face Incompressible Flow Regime	Public Domain Approximations			
	Debris Ingestion to HPC Trajectory	CFD + Kinetics	Meanline + Kinetics	Higher Order Flow Model + Kinetics		
	Crack Growth	Linear Elastic Fracture Mechanics (LEFM) da/dN vs. ΔK	Linear Elastic Plastic Fracture Mechanics			
	Life Cycle Cost	Acquisition	Implementation	Fuel Effciency Degradation	Repair/Replace Burdened Labor	Replace Material

4.2 FOD ON RUNWAY MODEL

4.2.1 FOD SIZES

As was mentioned before Foreign Object Debris is pervasive on runways [17, 30]. Hard Body FOD, a focus of this thesis, can range from small pebbles to large aircraft parts [17, 30]. Small metallic parts, rocks or pieces of the runway material (bitumen), less than 2g, pose a greater risk to the smaller HPC rotor blades than to the larger fan blades at the LPC on a jet engine – the fan blades have more area and material to absorb damage, whereas HPC rotors blades are smaller and thinner [17, 30]. A recent study at four military bases and two research stations in the UK revealed that 87.6 % of the debris collected (Hard Body FOD) on the runways was smaller than 5 mm in diameter [93] (see Table 4.4) – keep in mind that a 5 mm diameter steel sphere would only weigh 0.5g. NATO's technical report on mitigating damage to compressor blades presents spherical, metallic Hard Body FOD left on runways and flight decks that is smaller than 4 mm in diameter [17] (see Figure 4-1) – the report also evinces that Foreign Object Damage occurs beyond the LPC fan stages (see Figure 4-2). Despite ample evidence that small, Hard Body FOD poses serious economic risk, it takes a backseat to bird strikes – the current paradigm for airworthiness certification mandates structural robustness towards bird strikes [30].

Table 4.4: FOD Sizes from Runways at Military Bases and Research Stations

Size Fraction (g)	All(g)	%	Lyneham	Boscombe	Coltishall	Lossiemouth	Coningsby	Warton
% of Each			21%	7%	13%	25%	27%	7%
$d > 10$mm	1,086	7·2%	809	28	(Veg)	100	149	0
$10 > d > 5$mm	798	5·3%	227	54	19	324	174	0
$5 > d > 2$mm	3,368	22·2%	811	305	44	1,358	462	388
$2 > d > 1$mm	2,501	16·5%	519	270	231	859	419	203
$1 > d > 0{\cdot}5$mm	3,507	23·1%	328	243	1,165	757	730	284
$d < 0{\cdot}5$mm	3,925	25·8%	549	148	506	373	2,141	208
Total	15,185		3,243	1,048	1,965	3,771	4,075	1,082

Table 2
Size fractions (in grammes) from the different airfields

Figure 1: Typical Runway Foreign Object Debris.

Figure 2: Typical Runway Foreign Object Debris, as Picked up by FOD BOSS® System Runway Cleaning System.

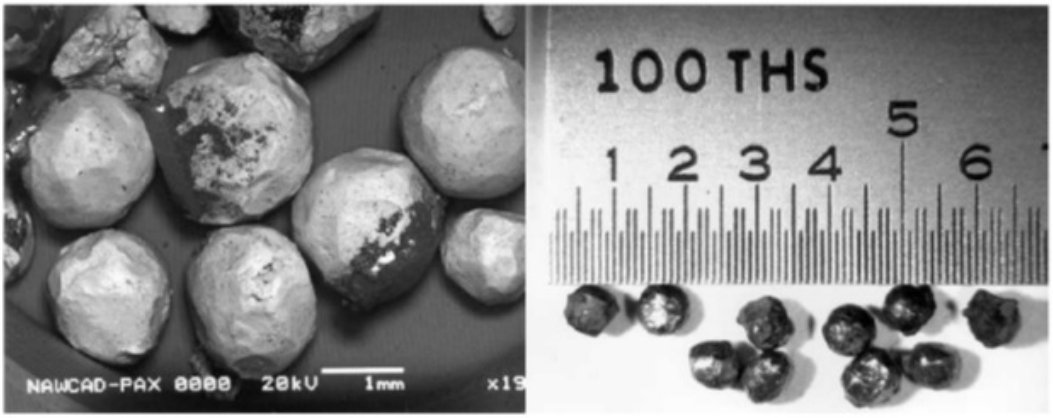


Figure 4: US Navy Foreign Object Debris – Small Shot Peen used to Clean and Repair Flight Deck (magnified views, units in right photograph are inches).

NATO RTO-TR-AVT-094

Figure 4.1: Typical Runway Debris, Military Runways

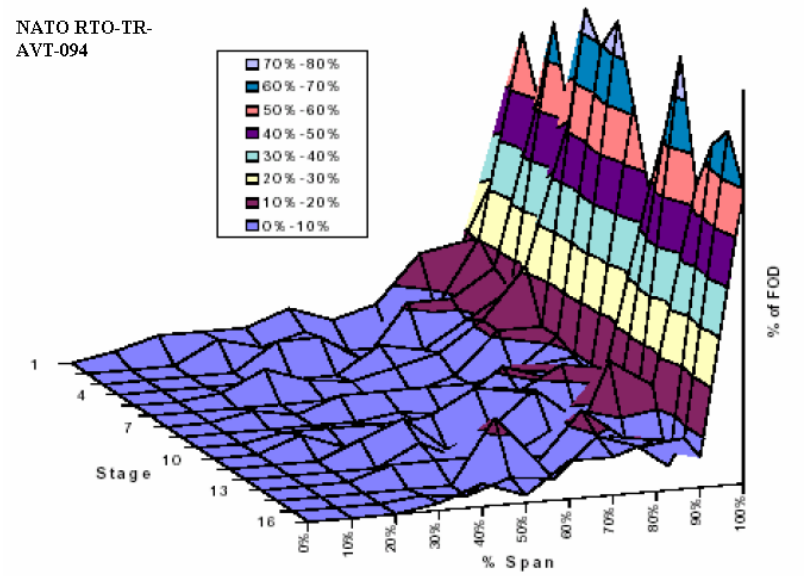


Figure 2: Percentage of FOD Located along the Span Relative to the Blade Tip.

Figure 4.2: Foreign Object Damage Engine Stage vs. % Rotor Span

4.2.2 RUNWAY DENSITY

An AIAA paper presents FOD runway density (amount of debris per runway) for military operations of fighters out of combat locations at less than 1 per 100 ft^2 (1 per 9.29 m^2) [94]. This is the density that will be modeled in this thesis – technologies and processes will operate on this density ρ_{runway}. For the runway considered in this thesis of 10,000 ft. length by 250 ft. width (3,048 m by 76.2 m) a total of 24,606 pieces of FOD less than 5 mm in diameter are calculated (1 per 9.29 m^2×3,048 m×76.2 m).

4.2.3 RUNWAY DISTRIBUTION

McCreary's extensive research into the impact of FOD to commercial operators evinces that debris on the runway is evenly distributed through-out the operational day [30]. With this assumption the model in Figure 4-3 is proposed for runway debris distribution. By means of particle to particle distance d Equations 6 and 7 can be related to reduce the number of variables to relate the counters for space between particles on periphery, respectively M and N (Equation 8). Equation 9 represents the total number of particles on the runway in relation to space counters M and N (Equation 9a reduces the formulation further).

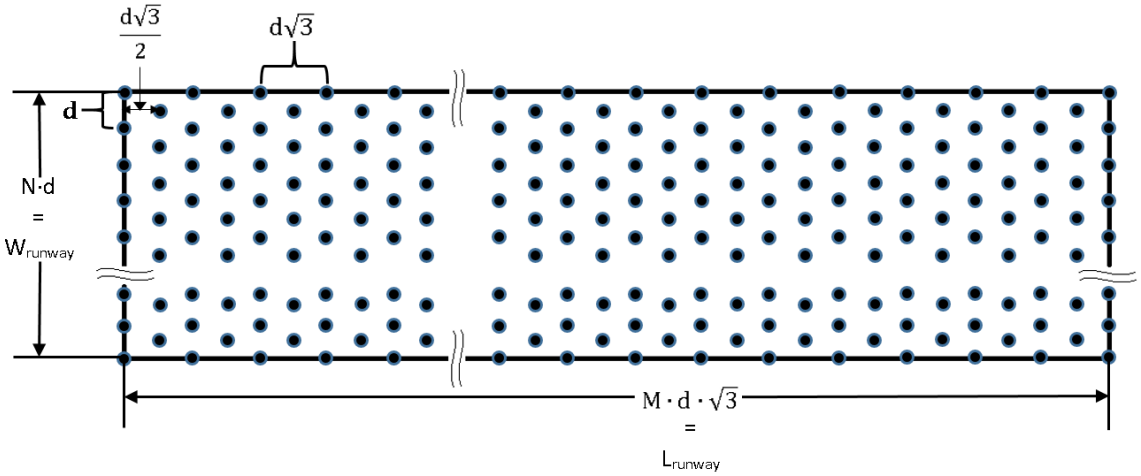


Figure 4.3: FOD on Runway Distribution Model

$$N \cdot d = W_{runway} \tag{6}$$

$$M \cdot d\sqrt{3} = L_{runway} \tag{7}$$

$$M = {}^{N \cdot L_{runway}}\Big/_{W_{runway} \cdot \sqrt{3}} \tag{8}$$

$$Total\ Debris\ on\ Runway = 2 \cdot (M+1) + 2 \cdot \{(N+1) - 2\} + (M-1+1) \cdot N + (M-2+1) \cdot (N-1) \tag{9}$$

After simplifying,

$$Total\ Debris\ on\ Runway = 2 \cdot M \cdot N + M + N + 1 \tag{9a}$$

Recall that total number of FOD particles on a runway is related to debris density,

$$Total\ Debris\ on\ Runway = \rho_{runway} \cdot L_{runway} \cdot W_{runway} \tag{9b}$$

Where,

$2\cdot(M+1)$ – number of debris particles on length of runway (L_{runway})

$2\cdot\{(N+1)-2\}$ – number of debris particles on width of runway (W_{runway})

$(M-1+1)\cdot N+(M-2+1)\cdot(N-1)$ – number of debris particles on inside of runway

ρ_{runway} – on runway FOD density

d – runway width-wise FOD particle pitch

4.3 GEOMETRY EXTERNAL TO ENGINE MODEL

4.3.1 ENGINE INLET DIAMETER

The engine to be modeled in this thesis is the Pratt & Whitney F100 engine – amply used by the F-15 and F-16. The choice of the engine was rather of convenience – some geometry and performance values are available in the public domain for the full line of these engines. Key to this thesis is the inlet diameter of the engine – the 0.88 m diameter of the F-100 will be used as full scale dimension of the modeled engine in this thesis [95].

4.3.2 Inlet Height to Diameter Ratio & Vortex Distance

The engine to be modeled for this thesis will place the face of a single LPC fan at the inlet diameter – the decision was made to not include cowling or inlet thickness structure as that information tends to be highly proprietary and not in the public domain. The distance from the ground plane to the lowest point on the inlet (often called the highlight plane) will be referred to as engine height, h. The ratio of the engine height to inlet diameter D_l (h/D_l) is a parameter that has a significant influence to the phenomena of intake ground vortex aerodynamics and debris aspiration via ground vortex action [96, 97] (see Figure 4-4). Figure 4-4 also presents two additional parameters that will be dialed to gage their effect on intake ground vortex aerodynamics and debris aspiration via ground vortex action – the horizontal and vertical distance from the vortex core on the ground to the location on the LPC fan face where it is ingested. In a later section the engine height to inlet diameter parameter h/D_l, horizontal/vertical distance from the ground vortex core to fan face will be dialed to gage their influence on debris aspiration.

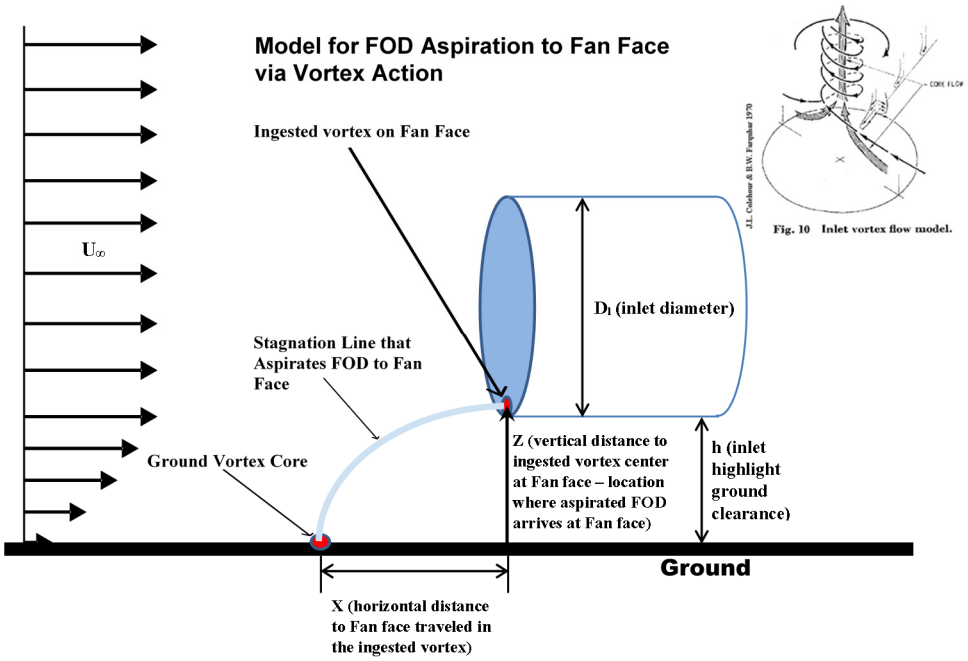


Figure 4.4: Ground to Fan Face FOD Aspiration Model

4.4 OPERATIONAL GROUND CONDITIONS

For this thesis runway operations will be synonymous with aircraft operation at the start of a runway during the early takeoff phase:

- Aircraft at beginning of takeoff phase on the runway
- Engine is throttled to 100% power setting – at this point the engine is at what is known in the operational vernacular as "High Suck, Low Speed" – the riskiest regime for aspiration of ground debris
- The aircraft begins to roll down the runway and for the next 3 – 4 seconds vortices will form from the fan face to the ground ahead, just below or behind the location on the ground where the engine inlet lies – the strength of the vortices will keep increasing with wind velocity, up to a peak, after which a further increase in velocity (in this case the wind velocity is a Head Wind) blows them away [96, 97] (see Figure 4-5)
- In the paper by Lee, Singh and Probert that modeled the F100-PW-229, the assumption was made that the inlet Mach number across operating conditions varied from 0.4 to 0.5, while Murphy in his Ph.D. thesis matched by dynamic similarity inlet Mach numbers for quiescent, headwind/rolling ground and headwind conditions of a $1/30^{th}$ scale inlet model in the rage of 0.29 to 0.58 [96, 98] – for this thesis the inlet Mach number will be varied from 0.45 to 0.5 and the modeled engine internal turbo machinery will be sized via a Meanline model in later section at an inlet Mach number of 0.475
- A Standard Day at Sea-level will be assumed [99]

Note: Dynamic similarity is a principle where if scale is varied between two geometrically identical replicas the full scale model independent and dependent non-dimensional parameters are equal to the scale model independent and dependent non-dimensional parameters respectively [102].

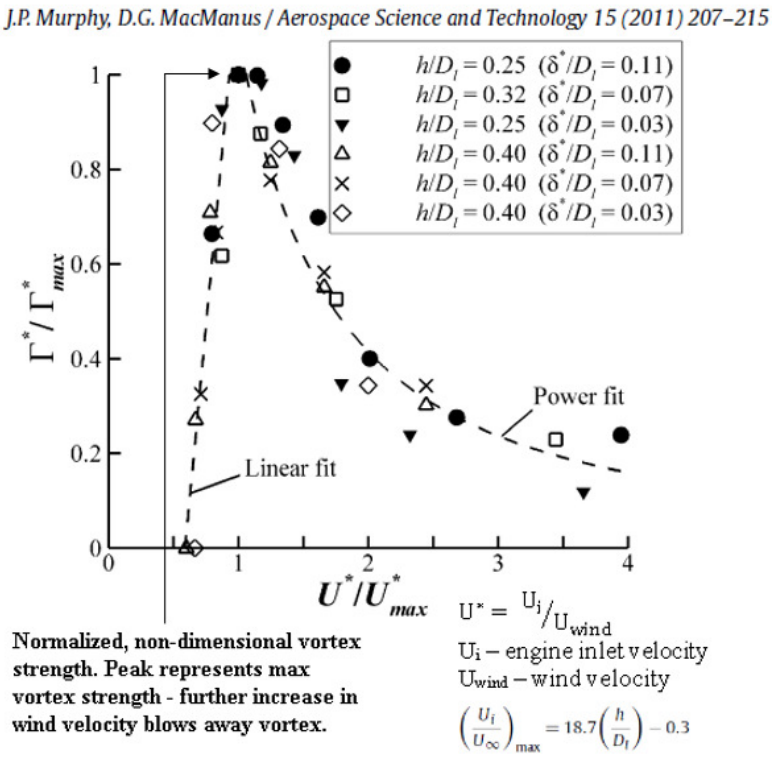


Figure 4.5: Normalized Non-Dimensional Vortex Strength

4.5 VORTEX MODEL

4.5.1 VORTEX FORMATION

A vortex describes the region in a fluid where the flow rotates about a line axis. During slow, ground operations vortices are known to form between the inlet of an aircraft engine and a stagnation point (a point where the local flow field is isentropically brought to rest, from the Bernoulli equation this point is where the static pressure is highest) on the ground (see Figure 4-6). The formation of the stagnation point on the ground is the catalyst

to the formation of the vortex from the ground (at the stagnation point) to the engine inlet – the stagnation point serves as the focal point of upstream vorticity that is concentrated and stretched to the intake [96, 97]. The stagnation point forms as a result of the interaction of the streamtube (the tube of airflow that contracts from a larger area upstream to a smaller area at the inlet) with the ground – the streamtube area upstream of the inlet is constrained by the ground, causing flow/ground interactions (see Figure 4-6) [96]. The parameters that dominate the phenomena of streamtube and ground interaction are the height to ground ratio (h/D_l) and the inlet to wind velocity ratio (U_i/U_∞, this ratio for incompressible flow is equal to the inverse of the area contraction ratio A_i/A_∞ that is a result of conservation of mass) [96, 97]. Murphy identifies the current trend for engine placements in his thesis [96] – with the least vortex generation risky geometry and wind velocity ratio H/D_l of and U_i/U_∞ of 0.875 and 12.75 respectively (on a Standard Day), Murphy's plot of engine placement trend evinces that for the geometries considered under this thesis vortices will occur between the stagnation point and the fan face (see Figure 4-7).

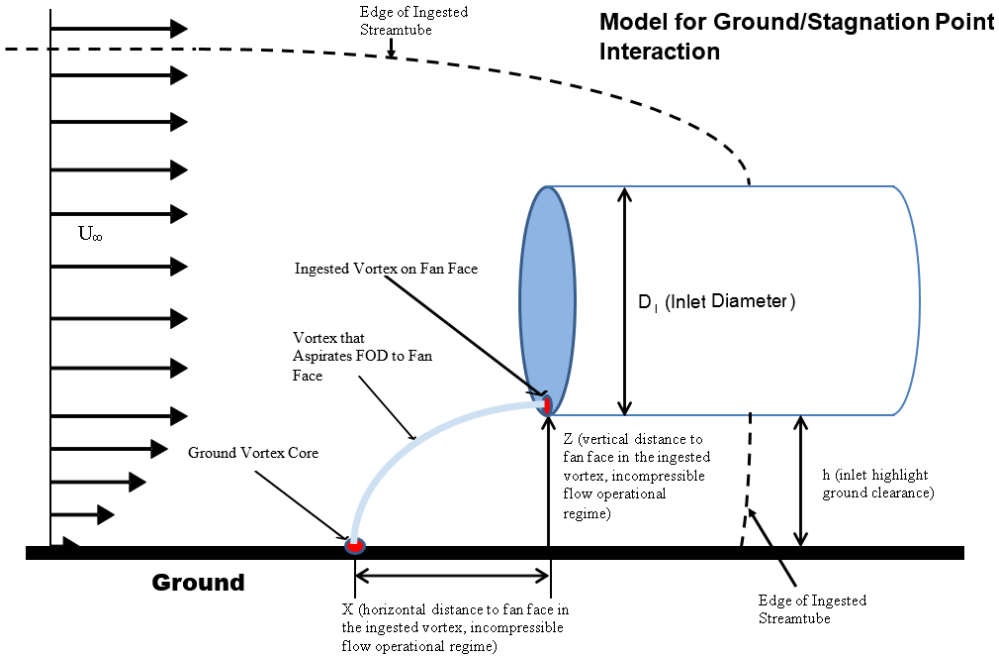


Figure 4.6: Model for Ground/Stagnation Point Interaction

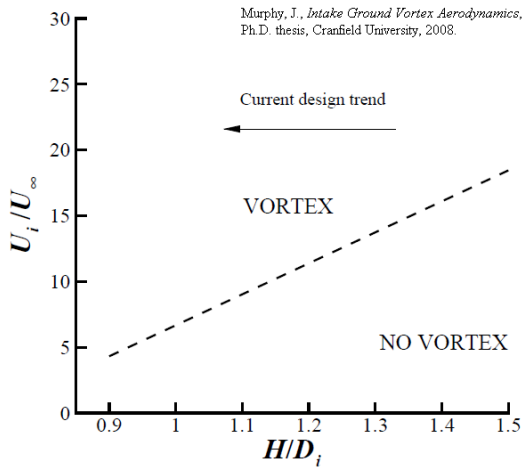


Figure 4.7: Engine Placement Trend

4.5.2 VORTEX STRENGTH & ASPIRATION CONDITIONS

Murphy's paper for vortex ingestion under headwind conditions provides a means to estimate the strength of the vortices that are generated under the operational and formation conditions already stated in this thesis, where Quiescent, Headwind is equal to Rolling Ground and Headwind conditions are considered [100] – Quiescent conditions are a subset of Headwind [96]. Under the assumption that any runway boundary layer effects are negligible, since the boundary layer at an airport is taller than the aircraft that operate on it, Figure 4-8 clearly demonstrates that for U^*/U^*_{max} of 3.9 (this non-dimensional vortex strength represents the smallest U_i/U_∞ (U^*) ratio of 18.9 and the weakest U^*_{max} of 4.8 calculated from Murphy's updated equation for that parameter [100]) the vortex present is strong, as it is ascending towards the maximum non-dimensional vortex strength with decreasing wind velocity.

The paper by Glenny and Pyestock on debris ingestion via vortex action (they use the term ingestion for the total action of aspiration from the ground to ingestion into the engine) presents thresholds for ingestion [97] (see Figure 4-9). Note that their thresholds

are for inlets much smaller than the one considered in this thesis, 12 in and 6 in vs. 0.88 m. In their paper they make the assertion that, "...because the Rossby number is inversely proportional to the inlet diameter it will be smaller for the larger intake. Thus on the evidence of Figures 9, 10 and 11, at a given (H/D), V_C/V_I will have to be increased to a higher value to blow the vortex away and so eliminate the risk of debris ingestion. This means that for the intake diameter (0.88 m) considered in this thesis and for H/D_I values of 0.77 to 0.875 (in a later section, h from the engine modeled in this thesis will be shown to vary in the design of experiments from 0.24 m to 0.33 m and for Glenny and Pyestock's paper H is the height to the center of the engine, i.e. $H = h + D_I/2$, where $D = D_I$) V_C/V_I would have to be increased considerably more than the cases considered by Glenny and Pyestock to blow away the vortex - evincing that debris ingestion for the engine considered in this thesis will occur. In their paper they also assert that, "In the scaling the necessity will be recalled for retaining the same Rossby number at both the model and full-scale, so that the prediction of maximum particle size just made necessarily assumes full-scale Froude and Rossby numbers similar to those employed in the model work." This assertion allows the author of this thesis to assume that the Rossby numbers Glenny and Pyestock used in their paper are not very different from those of today's engines – specially, since their paper focused on engines with large inlets low to the ground, just as this thesis does. If the highest H/Dl of 0.875 and the highest U_∞/U_i of 0.078 on a Standard Day (V_C/V_I in Glenny and Pyestock's paper for zero crosswind to engine centerline angle, effectively represents headwind conditions, a focus of this thesis)(see Figure 4-9) are considered along with the assertion that the Rossby numbers in the plot would not change drastically for a modern engine that is low to the ground and has a large intake defends the assumption that

ingestion is not just possible, but highly probable - the author of this thesis makes the assumption that based on these conditions and on runway operational conditions presented in Section 4.4 aspiration via vortex action will occur.

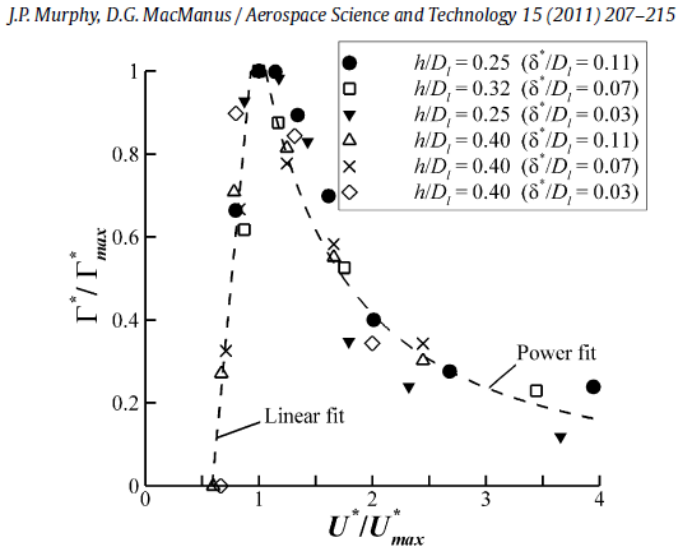


Figure 4.8: Normalized Non-Dimensional Vortex Strength

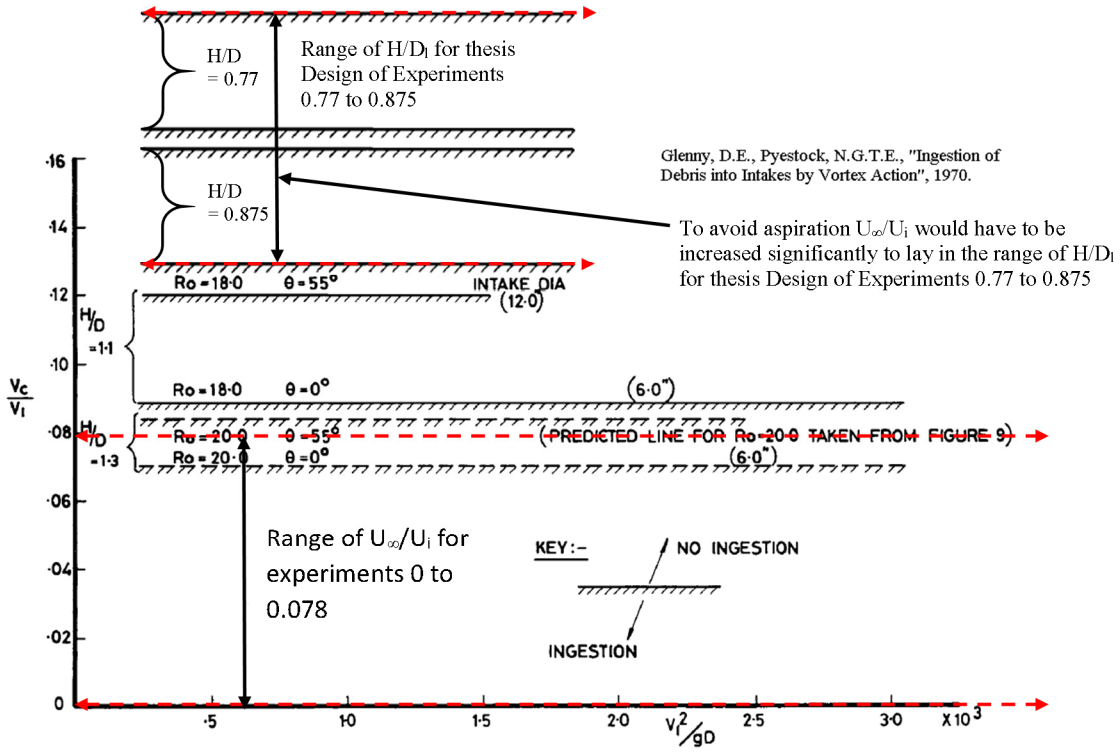


Figure 4.9: Ingestion Thresholds

4.5.3 VORTEX ON GROUND & FAN FACE MODELS & RESULTING FITS

Key to estimating the velocities of the FOD particles arriving at the engine fan face is the model for particle aspiration that will be presented in Section 4.6 – this model will depend on the pressure differences between ambient pressure and core pressure along the path from the ground to the fan face and any effect of the horizontal and vertical distance of the stagnation point to where on the fan face the vortex/particles are located/arrive (in this model the assumption is made that the particles arrive at the center of the vortex on the fan face).

The following data from Murphy's work on the aerodynamics of ground vortices presents will be leveraged in this thesis to estimate/model pressure ratios on the ground and fan face and the locations on the fan face of the ingested vortex/aspirated debris [96] – see Appendix A for full data set:

- h/D_l – height to inlet diameter ratio
- M_i – inlet Mach number
- U_i – inlet velocity, function of inlet static temperature and Mach number
- U_i/U_∞ - velocity ratio, the inverse of the area contraction ratio Ai/A_∞
- Non-dimensional boundary layer – a parameter used by Murphy in closed laboratory tests, for full scale engines operating from a runway the parameter is insignificant as the runway boundary layer is larger than the aircraft
- Γ^* - Non-dimensional vortex strength
- Fan face location of aspirated debris

4.5.3.1 Stagnation Point Pressure Ratio

A key parameter in modeling the aspiration of debris aspiration to the engine is the pressure ratio on the ground, at the stagnation point - that parameter was not directly considered by Murphy [96]. The author of this thesis proposes an estimate based on Murphy's Non-dimensional vortex strength (Γ^*) and vortex aerodynamics to estimate the pressure ratio at the stagnation point (go to Appendix A for the full data set and estimations). The following assumptions are made:

- The vortex core is circular in geometry
- The vortex on the ground can be divided into two regions [101]:
  - A rotational core, whose velocity profile is modeled by $V=\omega \cdot r$ (where r ranges from 0 at the center to the vortex core radius r_c and ω is the angular velocity that because of dynamic similarity will keep its values) and whose motion is that of a rigid rotating fluid
  - Beyond $r = r_c$ the velocity profile is assumed, irrotational and modeled by $V = C/r$, where C is a constant
  - At $r = r_c$ the tangential velocities must be the same:

    $\omega \cdot r_c = C/r_c$

    Leading to,

    $C = \omega \cdot {r_c}^2$ (10)
- For use on full scale geometry considered in this thesis dynamic similarity is also assumed, thus the stagnation pressure ratios will keep the values estimated with Murphy's data
- Pressures at both regions can be estimated from equilibrium – centripetal acceleration must be equal to pressure force (recovering Bernoulli's equation):

$$-\rho \cdot \frac{V^2}{r} = -\frac{dp}{dr} \tag{11}$$

At the rotational region,

$$\frac{dp}{dr} = \rho \cdot \frac{(\omega \cdot r)^2}{r} \tag{11a}$$

After integration,

$$p = \rho \cdot \frac{(\omega \cdot r)^2}{2} + A \tag{12}$$

At the irrotational region,

$$\frac{dp}{dr} = \rho \cdot \frac{({}^{\omega \cdot r_c{}^2}/_{r})^2}{r} \tag{13}$$

After integration (p at r = r_c to p_∞ at r = r_∞),

$$p = p_\infty - \rho \cdot \frac{(\omega \cdot r_c)^2}{2} \tag{14}$$

At r = r_c pressure must be equal, leading to the solution for constant A,

$$A = p_\infty - \rho \cdot (\omega \cdot r_c)^2$$

Thus the vortex pressure in cylindrical coordinates is,

$$p = p_\infty + \rho \cdot \frac{(\omega \cdot r)^2}{2} - \rho \cdot (\omega \cdot r_c)^2 \tag{15}$$

- An average pressure ratio at the stagnation point is thus estimated by the proposed formulae:

  a total force inside the vortex core is estimated as,

$$F_{total} = (p_\infty - \rho \cdot (\omega \cdot r_c)^2) \cdot \pi \cdot {r_c}^2 + \int_0^{r_c} \int_0^{2\pi} (\rho \cdot \frac{(\omega \cdot r)^2}{2}) \cdot r \cdot dr \cdot d\theta \tag{16}$$

  The pressure ratio at the stagnation point thus becomes,

$$\left({}^{p}/_{p_\infty}\right)_{Stagnation} = \frac{{}^{F_{total}}/_{\pi \cdot {r_c}^2}}{p_\infty} = 1 - \frac{{}^{3}/_{4} \cdot \rho \cdot (\omega \cdot r_c)^2}{p_\infty} \quad (17)$$

- The rotational velocity ω is not directly considered by Murphy, an estimate will be made from that data set [96]:
  - The rotational core of the vortex rotates as a rigid fluid, this idealization leads to vorticity, a measure of air rotation, being twice angular velocity
  - The vortex strength, also called the circulation (in the context of the vortex modeled in this thesis circulation is the line integral of velocity r·ω) can be calculated as follows:

    $$\Gamma = \oint_C \boldsymbol{V} \cdot ds = \int_0^{2\pi} (r \cdot \omega) \cdot r \cdot d\theta = 2 \cdot \pi \cdot r^2 \cdot \omega = 2 \cdot a \cdot \omega \quad (18)$$

  - Murphy presents non-dimensional vortex strength (average vortex strength $\bar{\Gamma}$ normalized by the product of inlet diameter D_l and inlet velocity U_i) Γ^*:

    $$\Gamma^* = \frac{\bar{\Gamma}}{D_l \cdot U_i} \quad (19)$$

    $\bar{\Gamma}$ is extracted by multiplying Γ^* by inlet diameter D_l and inlet velocity U_i,

    $$\bar{\Gamma} = \Gamma^* \cdot D_l \cdot U_i \quad (20)$$

  - With $\bar{\Gamma}$ at hand from Equation 20 and applying the relationship of Equation 18, ω can be calculated:

    $$\omega = \frac{\bar{\Gamma}}{2 \cdot a} = \frac{\Gamma^* \cdot D_l \cdot U_i}{2 \cdot \pi \cdot {r_c}^2} \quad (21)$$

4.5.3.2 Fan Face Location and Pressure Ratio

The vortex ingested at the fan face of the engine inlet modeled by Murphy has distinct locations and pressure ratios for the three regimes that are of interest to this thesis – quiescent (no wind condition on runway), headwind and headwind/rolling ground [96]

(see Appendix A and B for complete data set for all conditions). Quiescent conditions present the ingestion at the inlet of two counter-rotating vortices whose pressure ratios and vertical locations are influenced by the parameters h/D_l and M_i (see Figure 4-10) – the strongest pressure ratio and its height are recorded as the dominant ingested vortex. Headwind conditions present one ingested vortex at the inlet as soon as a small non-zero Head Wind velocity occurs (U_∞ is from a Head Wind) (see Figure 4-11) – the dominant parameters for this condition are U_i/U_∞, h/D_l, M_i and approaching boundary layer (has an insignificant influence in the context of this thesis because the runway boundary layer is taller than full scale aircraft). Headwind/rolling ground conditions present behaviors that range from no boundary layer, when the headwind and rolling ground velocity synchronize and eliminate the incoming vorticity that causes the boundary layer (effectively similar to quiescent conditions), to boundary layer behavior similar to the headwind conditions (see Figure 4-12). The dominant parameters for this condition are U_i/U_∞, h/D_l, M_i, approaching boundary layer and U_G (rolling ground). Data on pressure ratios and ingested vortex location was extracted with the aid of the color gradients on Murphy's plots and scaling rulers used to "size" how far from the inlet lip the vortices where located vertically [96] - the pressure ratio, based on dynamic similarity retains its magnitude (see Figure 4-10, Figure 4-11, Figure 4-12, Appendix A and B). Murphy's engine inlet model has a "lip", a transformation is made via Equation 23 to scale the vertical distance from the ground to the center of the ingested vortex on the inlet – the ingested vortex center is moved down by $(D_{scaled} - D_{l,scaled})/2$ to place it on the scaled inlet highlight diameter (see Figure 4-13). Appendix B contains the data and fit of the corrected vertical location of the ingested vortex on the inlet or fan face.

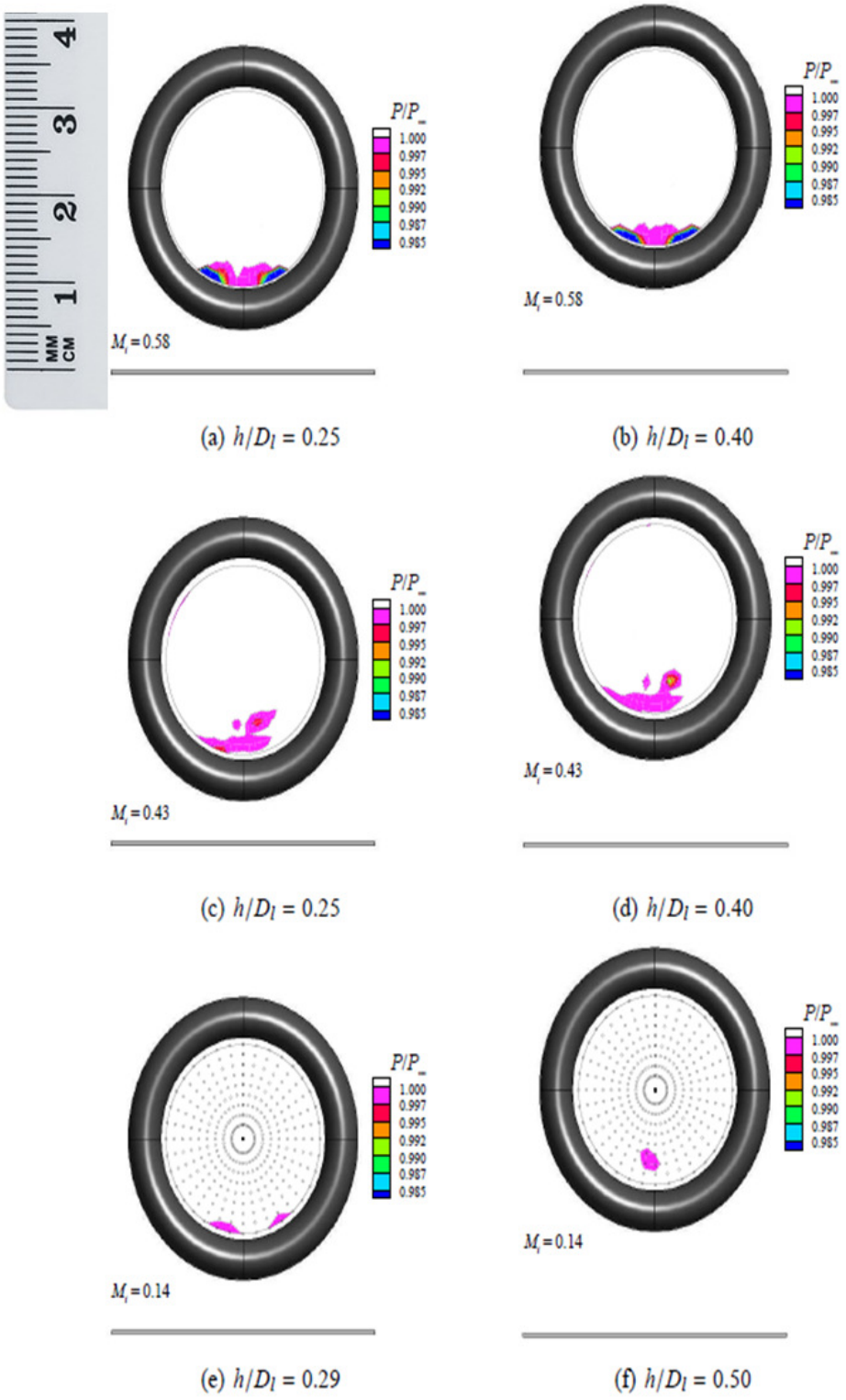


Figure 4.10: Effect of ground clearance, h/D_I, and intake Mach number, M_i on the total pressure contours within the intake duct under quiescent conditions Murphy, J.P., *Intake Ground Vortex Aerodynamics*, 2008

Figure 4.10: Quiescent Pressure Ratio Magnitude/Location

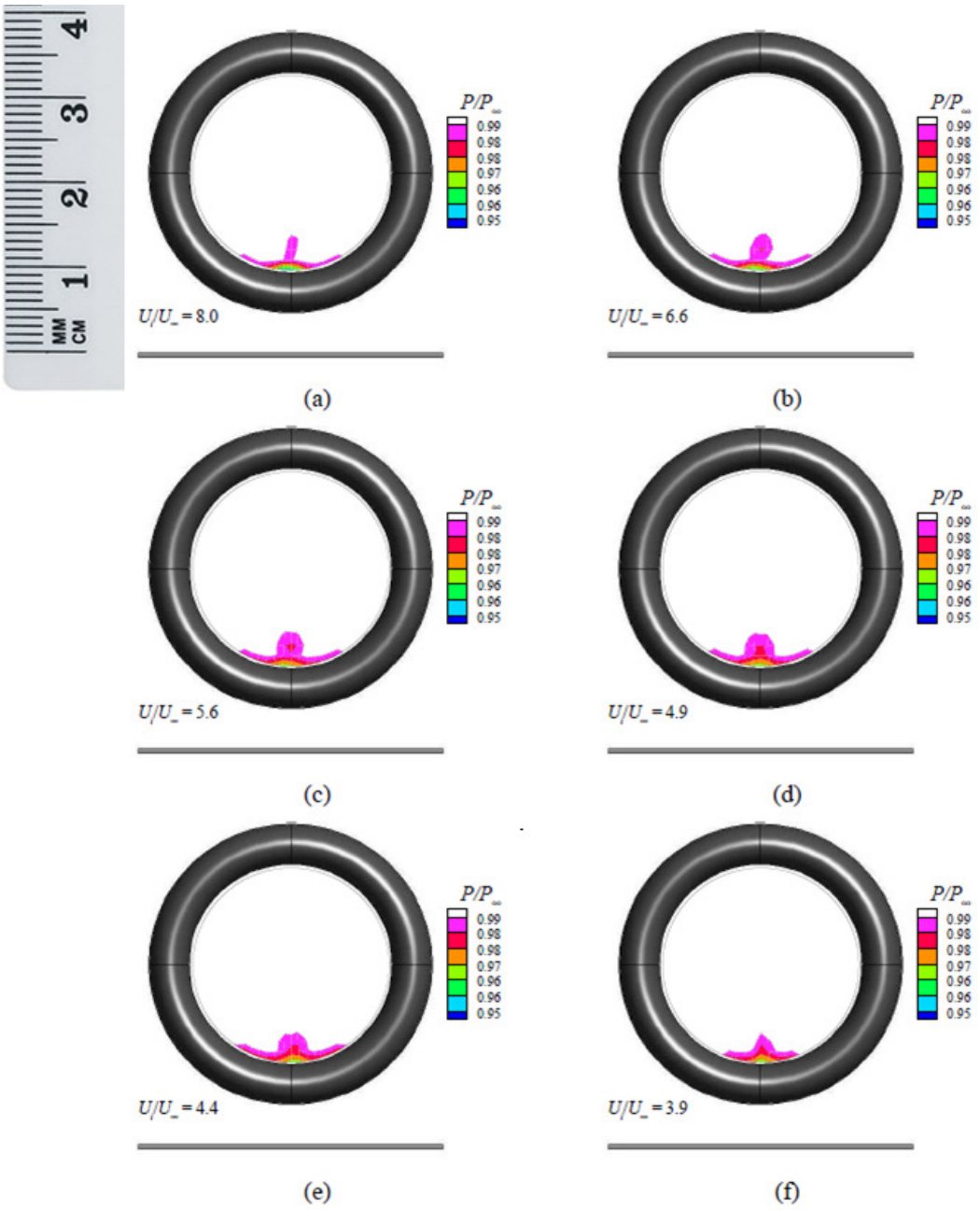


Figure 5.10: Effect of increasing headwind speed in the total pressure contours at the fan face for an $h/D_l = 0.25$, $M_i = 0.58$ and $\delta^*/D_l = 0.11$) Murphy, J.P., *Intake Ground Vortex Aerodynamics*, 2008

Figure 4.11: Head Wind Pressure Ratio Magnitude/Location

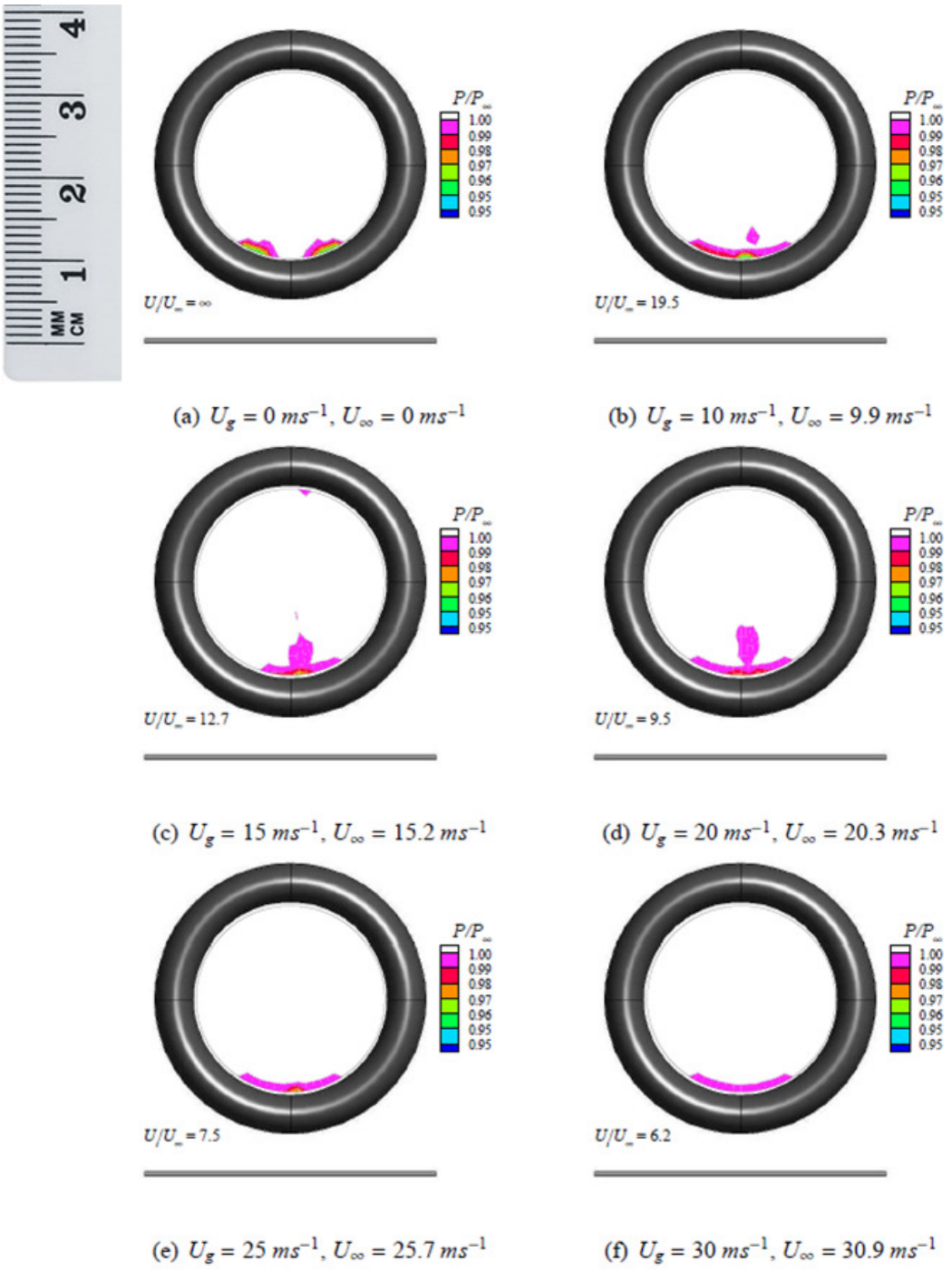


Figure 6.9: Contours of total pressure normalized by the working section total pressure for increasing ground speed from zero under no ambient wind conditions ($h/D_l = 0.25$, $M_i = 0.58$)

Murphy, J.P., *Intake Ground Vortex Aerodynamics*, 2008

Figure 4.12: Headwind/Rolling Ground Pressure Ratio Magnitude/Location

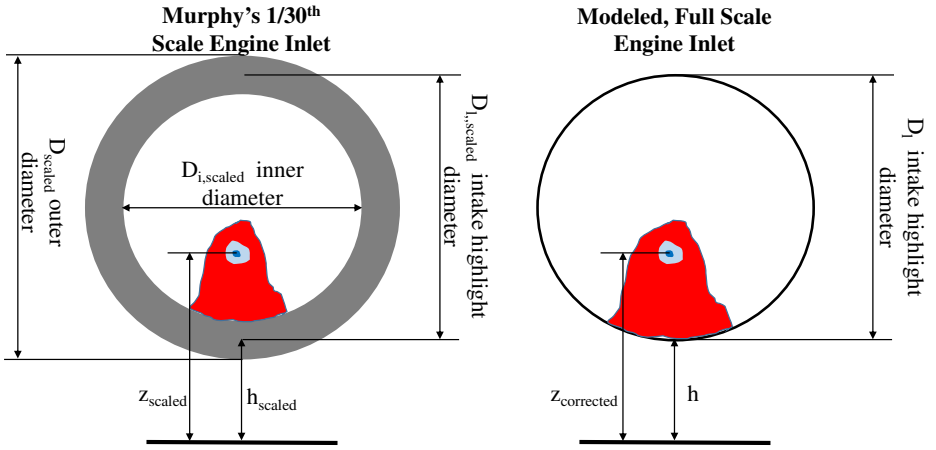


Figure 4.13: 1/30th Scale vs Full Scale Inlet Models

$$z_{corrected} = \left(\frac{z_{scaled} - \frac{D_{scaled} - D_{l,scaled}}{2}}{D_{l,scaled}} \right) \cdot D_l \tag{23}$$

4.5.3.2.1 Pressure Ratio & Fan Face Loc. Model Fits

The result of physically modeling the pressure ratios at the stagnation point and inlet and the full scale, corrected height of the ingested, dominant vortex center is a data set that is employed to generate fit formulas required in the particle aspiration model that will be presented in Section 4.6. The fits for the stagnation point and inlet/fan face pressure ratio were developed with Wolfram Mathematica® 10 while the fit for vertical location of ingested vortex was produced with JMP® Pro 11.2.0. Due to the limited training data from Murphy's experiments the goodness of fit tests were [30] (see Appendix A for raw data used to create models):

- R^2 (values greater than 0.90)
- Actual by predicted plots (whose data points are randomly scattered along a 45 degree line (perfect fit))
- Residual by predicted plots (whose data points present a random scattering with magnitudes much smaller compared to the predicted values)

- Model Fit Error (MFE, error of the model relative to actual values, at points employed to make the model)

The fits and statistical support are as follows:

- Pressure ratio at stagnation point

$$\left(P/P_{\infty}\right)_{stagnation} = AA \cdot \frac{h}{D_l} \cdot M_i + CC \cdot \frac{h}{D_l} \cdot U_G + EE \cdot M_i \cdot U_\infty + FF \cdot M_i \cdot U_G + LL \cdot h/D_l + MM \cdot {M_i}^2 + NN \cdot M_i + PP \cdot U_\infty + RR \cdot U_G + UU \quad (24)$$

Figure 4-14 presents goodness of fit measures and coefficients of fit,

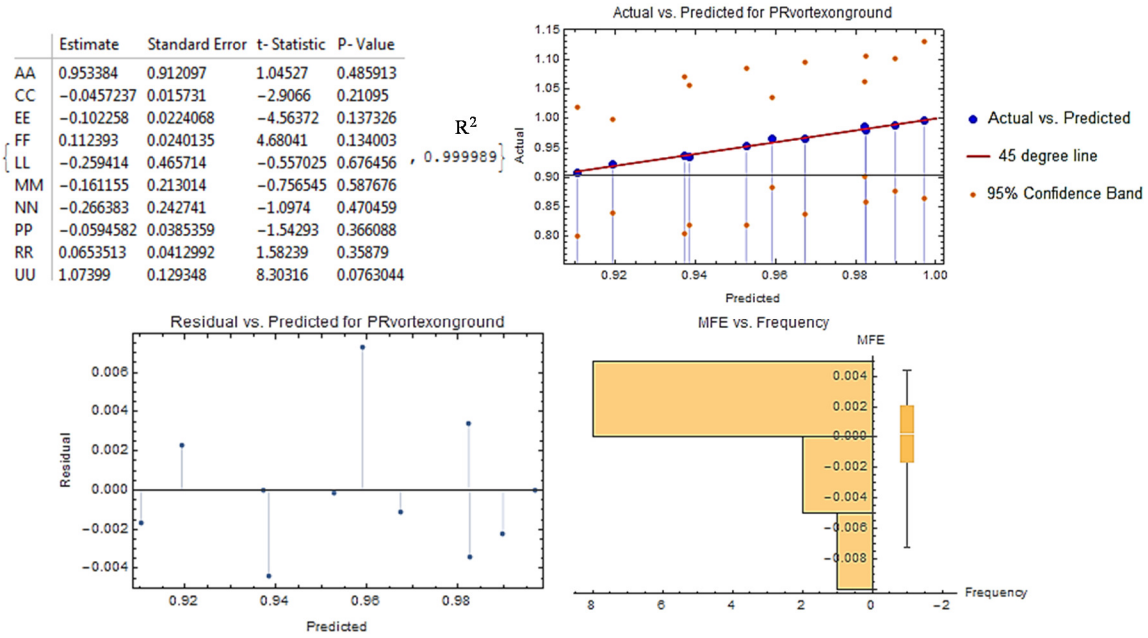


Figure 4.14: Stagnation Point Pressure Ratio Fit Measures

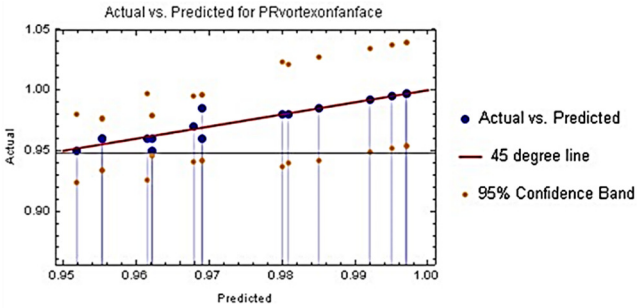
- Pressure ratio at inlet/fan face ingestion

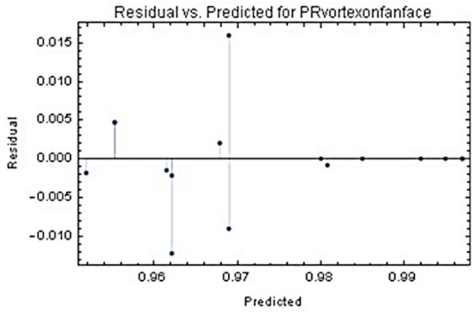
$$\left(P/P_{\infty}\right)_{inlet} = AA \cdot \frac{h}{D_l}^2 + BB \cdot \frac{h}{D_l} \cdot M_i + CC \cdot \frac{h}{D_l} \cdot U_\infty + DD \cdot \frac{h}{D_l} \cdot U_G + EE \cdot M_i^2 + FF \cdot M_i \cdot U_\infty + GG \cdot M_i \cdot U_G + HH \cdot U_\infty \cdot U_G + II \cdot \frac{h}{D_l} + JJ \cdot M_i + KK \cdot U_\infty + LL \cdot U_G + MM \quad (25)$$

Figure 4-15 presents goodness of fit measures and coefficients of fit,

	Estimate	Standard Error	t-Statistic	P-Value
AA	1.8723	3.08253	0.607392	0.586481
BB	0.834904	1.09139	0.764988	0.499951
CC	-0.00305058	0.00113322	-2.69196	0.0742906
DD	-0.00177313	0.000447291	-3.96415	0.0286756
EE	-0.226516	0.262476	-0.862997	0.451612
FF	0.00138654	0.00867179	0.15989	0.883126
GG	-0.00076237	0.00104031	-0.732827	0.516731
HH	0.00013047	0.000139751	0.933589	0.419384
II	-1.596	2.56571	-0.622052	0.577981
JJ	-0.15107	0.410232	-0.368256	0.737133
KK	-0.000762645	0.00453288	-0.168247	0.877092
LL	-0.000443282	0.00178916	-0.247759	0.82031
MM	1.29407	0.489402	2.64419	0.0773775

$\{R^2, 0.999964\}$





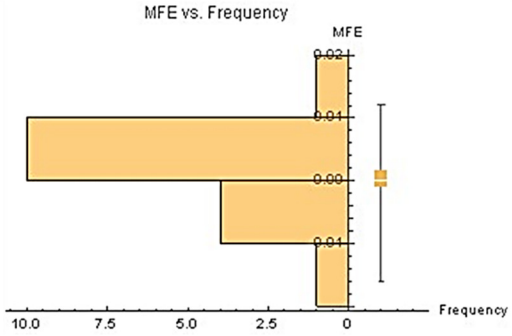


Figure 4.15: Inlet Ingestion Point Pressure Ratio Fit Measures

- Vortex ingestion vertical location at inlet/fan face

$$z_{inlet} = 0.40041 + 0.99052 \cdot h/D_l + -0.22183 \cdot \delta/D_l + -0.66479 \cdot M_i + 0.00027 \cdot U_\infty + -0.00179 \cdot U_G \quad (26)$$

Figure 4-16 presents fit quality measures,

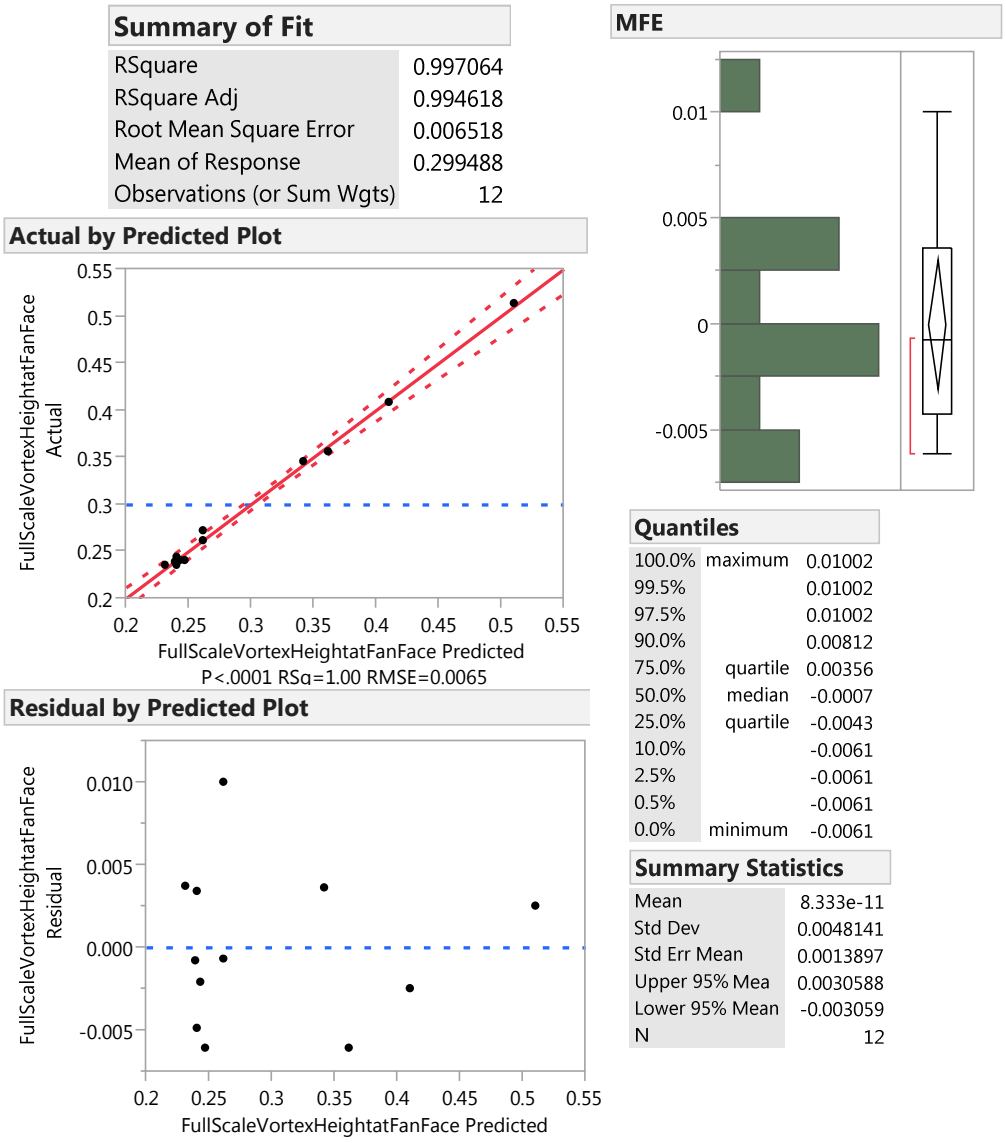


Figure 4.16: Inlet Ingestion Point Vertical Location Fit Measures

4.6 ASPIRATION MODEL

The conditions for vortex generation, strength and aspiration as well as the physical models for pressure ratios and location of ingested vortex at inlet/fan face have been established in Section 4.5. The following section will present the assumptions, conditions and physical representations required to model:

- Likelihood the ground vortex core (stagnation point) will move over a FOD particle sitting on the runway
- Forces that lift the particle off runway and aspire it inside the vortex to inlet
- Arrival velocities at the fan face location where the vortex is ingested – recall that the assumption is made that the aspirated debris particle arrives at the center of the ingested vortex at the fan face

4.6.1 Probability of Ground Vortex Core Engulfing Debris

The probability that a FOD particle on the runway is engulfed by the ground vortex core (a pre-requisite to aspiration, as the particle has to be inside vortex core to be aspirated off the ground) is a function of where on the runway the vortex core is during the early takeoff phase – at that phase strong vortices that aspire debris occur (see Section 4.5). The assumption is made that width-wise on the runway the vortex core has equal probability of being anywhere (uniform distribution) as the early takeoff phase begins – also assumed is that width-wise as the aircraft rolls down the runway it stays in the same trajectory, takeoff roll does not seem to have much "wiggle". Sections 4.2.2 and 4.2.3 provide the density and distribution of debris that will be considered in this thesis. Figure 4-17 presents the geometric parameters employed in the proposed equation (see Equation 27) to estimate the probability that a particle will be engulfed by the ground vortex core of radius r_c.

$$P_{EngulfedFOD} = \frac{N \cdot \left\{ D_{VortexCore} - \frac{D_{FOD}}{2} \right\}}{W_{runway}} \tag{27}$$

Where:

N – runway width space counter between debris particles (see Section 4.2.3)

$D_{VortexCore}$ – twice the radius of the vortex core, r_c

D – diameter of the FOD particle

W_{runway} – width of runway

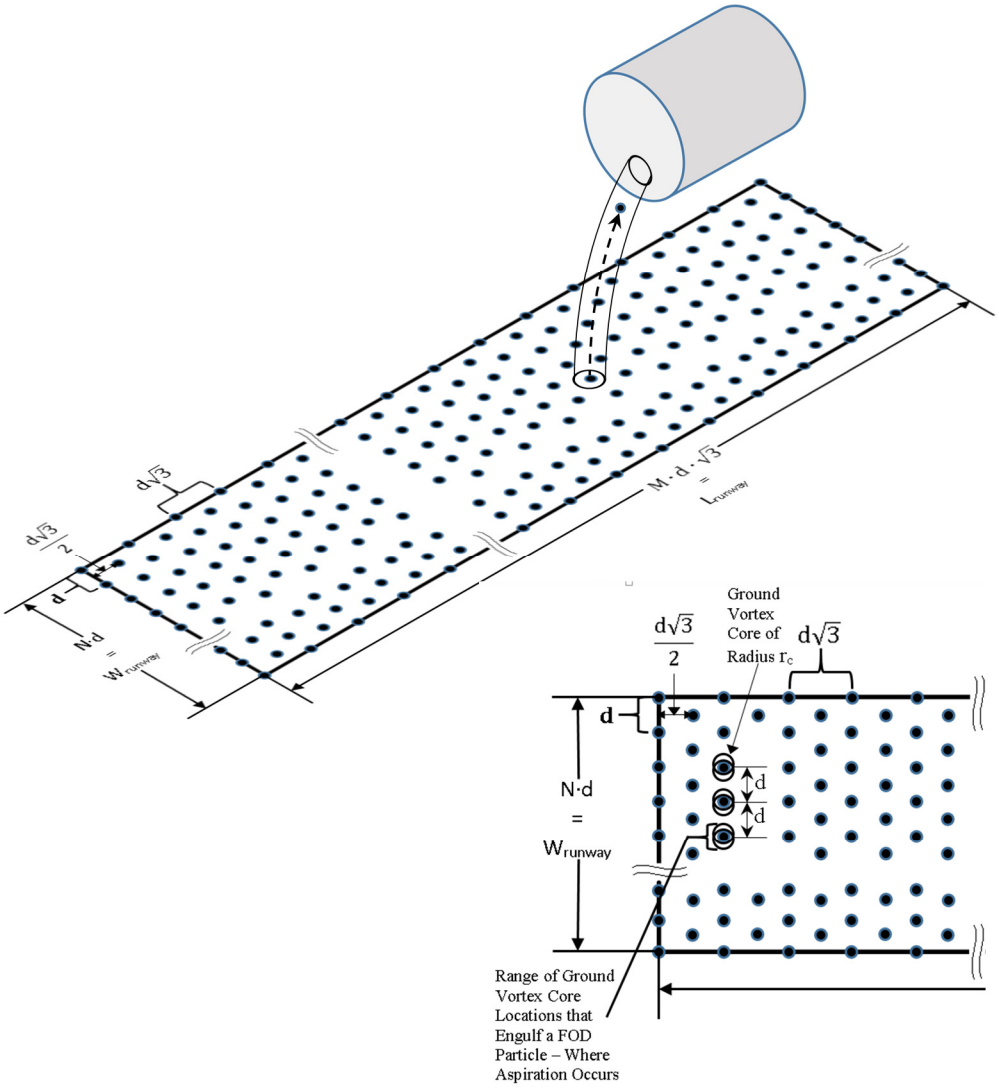


Figure 4.17: Probability that Ground Vortex Core Engulfs Particle

4.6.2 Aspiration Forces & Velocities

To derive a model to estimate the aspiration forces and velocities on a FOD particle that is engulfed by the vortex core and is subjected to the net impulsive resulting from the pressure ratios calculated under conditions set forth in Section 4.5 for vortex formation/strength and aspiration thresholds the following assumptions are made:

- From ground to fan face in the early takeoff phase air flow is considered incompressible – the maximum headwind condition that will be considered is 12 m/s, only at the inlet will Mach number vary between 0.45 and 0.5
- Frictionless/adiabatic flow – an isentropic process
- Flow is along a Streamtube – a defining condition for a vortex
- Flow is uniform at each section of the vortex
- By Bernoulli's equation (derived from equilibrium – acceleration must equal pressure force) static pressure, dynamic pressure and head (height pressure potential) are held constant for incompressible flow (see Equation 28) – the assumption is made that the dominant impulsive force that aspirates the FOD particle is caused by the pressure differential between the pressure inside the vortex and ambient pressure outside the vortex along with a net head, for this particular case total and static pressure are the same (see Equation 29) – the assumption is supported by tornado aerodynamics where inside the core axial velocities are small and yet aspiration of heavy objects occurs [103]

$$P = P_{static} + 1/2 \cdot \rho \cdot V^2 + \rho \cdot g \cdot z \quad (28)$$

$$P(\mathrm{z}) = P_{inlet} + \rho \cdot g \cdot \{z_{corrected} - z\} \quad (29)$$

Where,

ρ – flow density

V – flow velocity

g – gravitational acceleration

$z_{corrected}$ – height from datum (ground) to vortex core at inlet (Section 4.5.3)

z – height from datum (ground)

P_{inlet} – pressure inside the vortex core where ingestion at inlet occurs

Figure 4.18 presents the velocity and force model on a FOD particle aspirated from the ground to the inlet/fan face – the figure also presents visually the assumption of a parabolic particle trajectory inside the vortex during aspiration, this assumption is supported by scale and full size aspiration modes (see Figures 4-19 and 4-20)[97, 104].

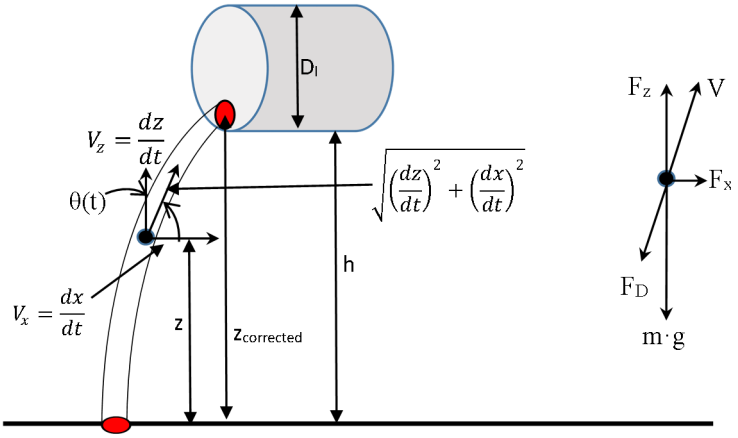


Figure 4.18: FOD Particle Aspiration Velocity/Force Model

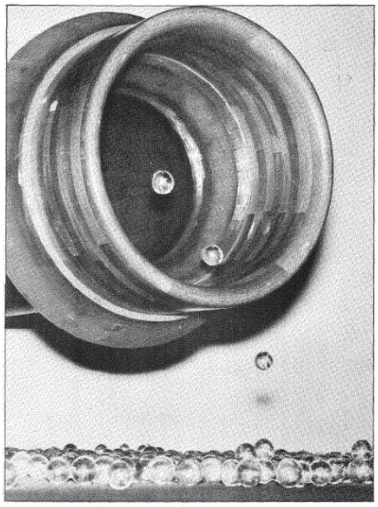
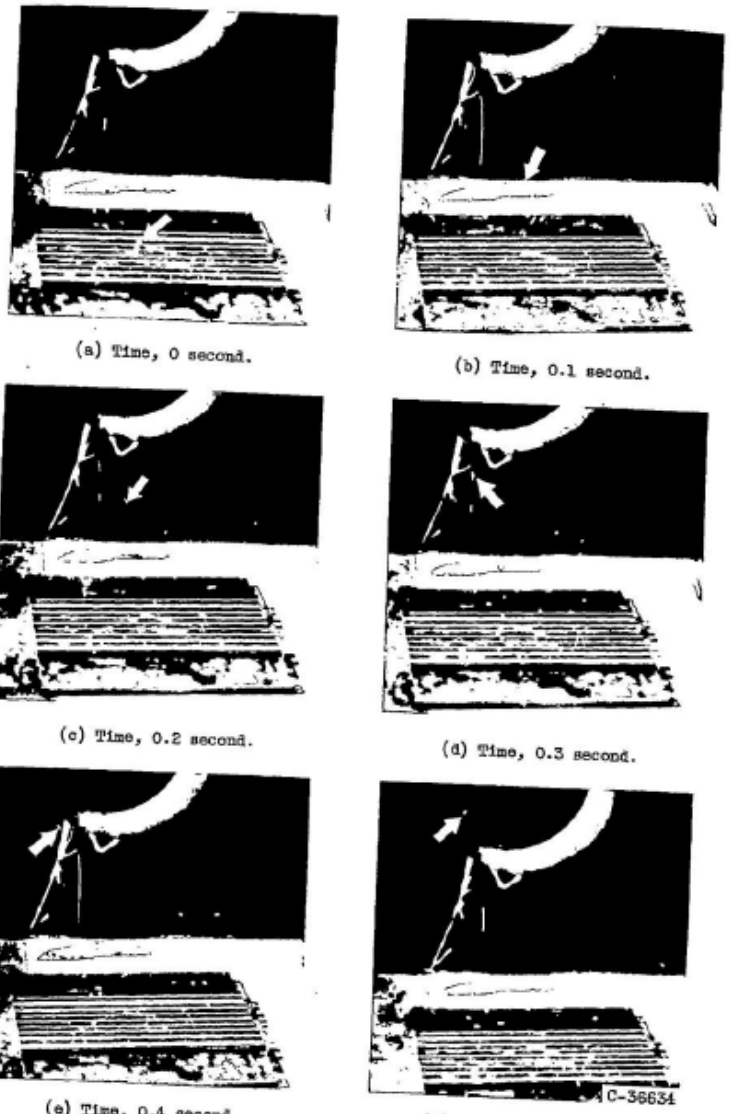


FIG.6 12in DIAMETER INTAKE AT AN H/D= 1.2 SHOWING INGESTION OF 1in. DIAMETER GLASS BALLS.
Glenny, D.E., Pyestock, N.G.T.E., "Ingestion of Debris into Intakes by Vortex Action", 1970.

Figure 4.19: Ingestion of Glass Beads into Diameter Intake



(a) Time, 0 second. (b) Time, 0.1 second. (c) Time, 0.2 second. (d) Time, 0.3 second. (e) Time, 0.4 second. (f) Time, 0.6 second.

Figure 11. - Path of a pebble from platform surface, shown in series of motion-picture photographs.

Figure 4.20: Particle Trajectory into Full Size Engine Test

The pressure ratios developed in Section 4.5 are employed via a linear relationship to estimate the net pressure on a FOD particle inside the vortex (see Equation 31). This net pressure along with drag and weight comprise the force system in the x-z plane (see Equations 32 and 33) – in this thesis crosswind conditions are not considered and the

trajectory plane of the FOD particle is 2-D and has an incidence of 0˚ with respect to the engine axis. The formulation is as follows:

- Stagnation point to vortex core ingestion point at inlet/fan face,

$$PressureSlope = \frac{P(z=z_{corrected}) - P(z=0)}{z_{corrected} - 0} = \frac{\left(P/P_\infty\right)_{inlet} \cdot P_\infty + \rho \cdot g \cdot z_{corrected} - \left(P/P_\infty\right)_{stagnation} \cdot P_\infty}{z_{corrected}} \quad (30)$$

Equation 29 via Equation 30 is converted to,

$$P(z) = \left\{\left[\left(P/P_\infty\right)_{inlet} - \left(P/P_\infty\right)_{stagnation}\right] \cdot z/z_{corrected} + \left(P/P_\infty\right)_{stagnation}\right\} \cdot P_\infty + \rho \cdot g \cdot z \quad (31)$$

- Forces on the particle,

$$\sum F_x: \left(P_\infty - P(z)\right) \cdot A_{FOD} \cdot \sin\theta(t) - \frac{1}{2} \cdot \rho \cdot {V_x}^2 \cdot C_D \cdot A_{FOD} = m \cdot \frac{dV_x}{dt} \quad (32)$$

Recall that,

$V_x = dx/dt$

$U_x = x$

$$\sin\theta(t) = \frac{dx}{\sqrt{(dx)^2 + (dz)^2}} = \frac{dx/dt}{\sqrt{(dx/dt)^2 + (dz/dt)^2}}$$

$$\sum F_x:$$

$$\left(P_\infty - \left\{\left[\left(P/P_\infty\right)_{inlet} - \left(P/P_\infty\right)_{stagnation}\right] \cdot \frac{z}{z_{corrected}} + \left(P/P_\infty\right)_{stagnation}\right\} \cdot P_\infty - \rho \cdot g \cdot z\right) \cdot A_{FOD} \cdot \frac{\frac{dx}{dt}}{\sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2}} - \frac{1}{2} \cdot \rho \cdot \left(\frac{dx}{dt}\right)^2 \cdot C_D \cdot A_{FOD} = m \cdot \frac{d^2x}{dt^2} \quad (32a)$$

$\sum F_z$:

$$\left(P_\infty - P(z)\right) \cdot A_{FOD} \cdot \cos\theta(t) - m \cdot g - \frac{1}{2} \cdot \rho \cdot {V_z}^2 \cdot C_D \cdot A_{FOD} = m \cdot \frac{dV_z}{dt} \quad (33)$$

Recall that,

$V_z = dz/dt$

$U_z = z$

$$\cos\theta(t) = \frac{dz}{\sqrt{(dx)^2 + (dz)^2}} = \frac{dz/dt}{\sqrt{(dx/dt)^2 + (dz/dt)^2}}$$

$$\sum F_z: \left(P_\infty - \left\{\left[\left({}^{P}/_{P_\infty}\right)_{inlet} - \left({}^{P}/_{P_\infty}\right)_{stagnation}\right] \cdot \frac{z}{z_{corrected}} + \left({}^{P}/_{P_\infty}\right)_{stagnation}\right\} \cdot P_\infty - \rho \cdot g \cdot z\right) \cdot A_{FOD} \cdot \frac{\frac{dz}{dt}}{\sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2}} -$$

$$m \cdot g - \frac{1}{2} \cdot \rho \cdot \left(\frac{dz}{dt}\right)^2 \cdot C_D \cdot A_{FOD} = m \cdot \frac{d^2z}{dt^2} \quad (33a)$$

The vortex ingestion/particle aspiration vertical location at inlet/fan face developed from scaled/corrected data from Murphy's work [96] along with setting the displacement at start to zero and the horizontal distance from stagnation point to fan face set via DOE setting provide the initial conditions.

4.6.3 Aspiration Simulation

The force system developed in Section 4.6.2 is a set of coupled differential equations that are solved numerically in the same Wolfram Mathematica® 10 code that was employed to develop the pressure ratio fits of Section 4.5.3.2.1. Once the condition for particle presence inside the stagnation point vortex core is met the solution to the force system provides the trajectory and velocity profiles from the ground to fan face. The assumption is made that because the engine modeled in this thesis did not include an inlet

that recesses the fan face, where effectively the vortex is turned horizontal as it reaches the fan face, only the horizontal component of velocity is considered at the inlet – this assumption is supported visually by public domain photographs of vortices ingested by jet engines where there may be a trend for straighter vortex tubes at lower engine heights, thus if the fan face is near the inlet the ingestion velocity would be mostly the horizontal component (see Figures 4-21 and 4-23). Note that the author of this thesis did not encounter studies that related fan face recess to vortex or FOD ingestion. Figures 4-24 and 4-25 provide typical trajectory and velocity profiles for the two steel particle sizes (1.33 mm and 3.2 mm diameter) whose case studies will be presented in a later section in this thesis. Appendix C presents the code, initial conditions for numerical solution and a discussion on how the code works.

Figure 4.21: F-16 Engine Ingesting a Vortex Streamtube

Rolls-Royce RB211-524G ©Peter Thomas 2005

Figure 4.22: 747 Engine Ingesting a Vortex

Figure 4.23: Boeing YC-14 Engine Ingesting Vortex

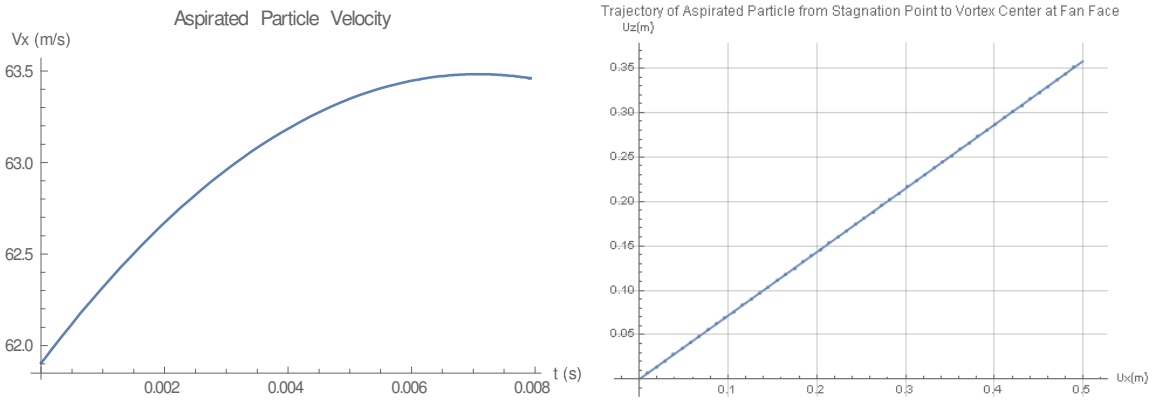


Figure 4.24: Aspiration Model Typical Velocity/Trajectory Profile 1.33 mm Particle

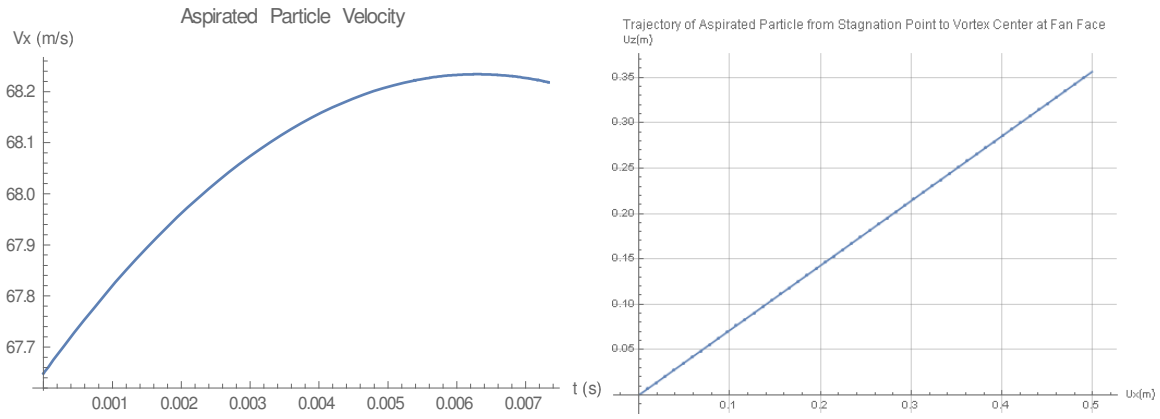


Figure 4.25: Aspiration Model Typical Velocity/Trajectory Profile 3.2 mm Particle

4.7 ENGINE MEANLINE ANALYSIS MODEL

The turbomachinery modeled for this thesis includes rotors, stators and guide vanes – on current engines the rotors are attached to rotor disks – the LPC fan is attached to a rotor disk just as the HPC rotor is. Rotors exchange kinetic energy for pressure rises – the blades do work on the flow they move across them. Stators and guide vanes condition or "straighten" and diffuse the flow before it enters the next rotor or exits to another component.

Fully 3-D flow dynamics and thermodynamics inside an operating jet engine are computationally intensive, an approximation is required to estimate the nature of the flow that a particle will encounter on its trajectory to the blades on the HPC rotor disk more efficiently. Additionally, the approximation must allow greater control of input variables as a range of conditions via a Design of Experiments (DOE) will be considered. A lower level approximation that provides ample information on the nature of the flow, allows sufficient control of the inputs, is not proprietary and is computationally economic (as opposed to full 3-D CFD codes that are complex and computationally intensive) is a Meanline Analysis. A Meanline Analysis, also called a Mean Streamline Analysis is a

streamline or 1-D first level principles approach to model the thermodynamics and flow dynamics inside an axial compressor – the approximation is effectively done across a mean or averaged line radially located from the shaft and chord-wise located from the leading to the trailing edge of the turbomachinery airfoil [105, 106]. Most of the current jet engine designs are of the axial compressor type – axial compressors compress and move flow through the action of rotating airfoils – the term axial is apropos because flow is moved with negligible incidence with respect to the axis of rotation.

A Meanline Analysis tool, spreadsheet-based and supported by Falck's work on the topic, was developed to provide 1st level principles estimates of [105]:

- LPC fan to HPC rotor static and total flow dynamics and thermodynamics
- A baseline sizing of the modeled engine turbomachinery – the Meanline Analysis tool was set to the average inlet Mach number of 0.475 and it provided the baseline, internal sizing to the turbomachinery: span (height from shaft), blade chord to pitch ratio (solidity), angles of the blades and chord lengths

Recall that the Pratt & Whitney F-100 low-bypass turbofan is being modeled for this thesis. To expedite the modeling of FOD particles from ground to impact at the 1st row of blades (a row of blades from a rotor disk is called a cascade) on the HPC rotor disk this set of engine sections will be simplified – the sections to be considered will consist of a single LPC fan at the inlet, a single stator behind the fan, a single Inlet Guide Vane (IGV) behind the stator and a single HPC rotor behind the IGV (see Figure 4-26). Available, public domain and simulated performance and geometric information is presented in Table 4.5 – as most of the physical engine's performance and configuration information is

proprietary a mix of Pratt & Whitney and third party openly available data had to be collected to model a "similar-enough" engine for this thesis.

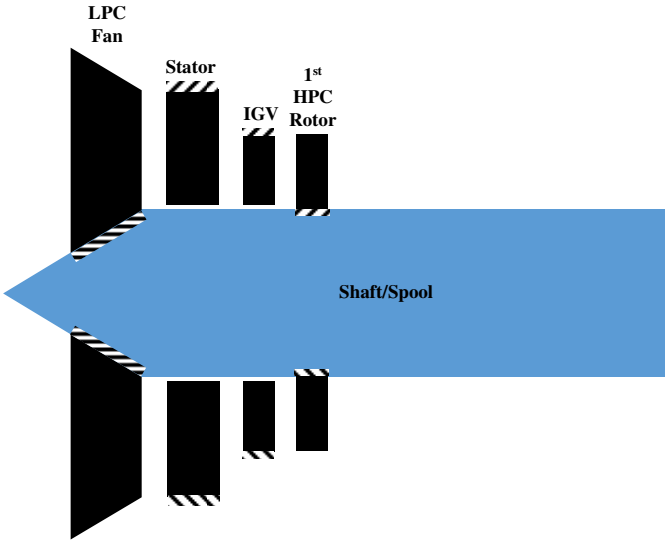


Figure 4.26: Modeled Engine Sections

Table 4.5: Engine Public Domain Information

Number of LPC Fan Blades (source: government Pre-solicitation Notice) [107]	Number of Stator Blades (source: author assumtion)	Number of IGV Blades (source: author assumption)	Number of 1st HPC Rotor Blades (source: author assumption)	Inlet Diameter (m) (source: P&W) [95]	Shaft Diameter (m) (author approximation)	LPC Fan Stages (source: P&W) [108]
38	38	34	34	0.88	0.42	3
HPC Rotor Stages (source: simulated engine) [98]	**Mass Flow Rate (kg/s) (source: simulated engine) [98]**	**Bypass Ratio (source: P&W) [95]**	**Overall Pressure Ratio (OPR) (source: P&W) [95]**	**OPR Fan Stages (source: P&W) (source: simulated engine) [98]**	**LPC Spool Angular Velocity, N1 (rpm) (source: NASA) [109]**	**HPC Spool Angular Velocity, N2 (rpm) (source: NASA) [109]**
11	112.7	0.36	32.4	3.8	10000	14000

4.7.1 MEANLINE ANALYSIS LINGO

A Meanline Analysis is a 1st level principles approximation for the flow dynamics and thermodynamics inside a turbomachine. Before the dynamics and thermodynamics methodology and formulation is presented a basic understanding of terminology is required. The following definitions along Figures 4-27, 4-28 and 4-29 serve as a guide:

- Meanline or Streamline – in the context of analysis of the aero/thermo dynamics of an engine refers to a line that starts at the inlet and continues aft and whose distance from the engine axis datum (center) is the root mean squared of the tips and hubs of the inlet and outlet respectively of a rotor, stator or guide vane
- Flow velocity triangles – at any point inside an engine the flow can be decomposed into its tangential, radial and axial components – Figure 4-28 presents in the context location and airfoil geometry the velocity triangle components with their definitions
- Particle trajectory in the context of geometry and velocity triangles – as an ingested particle travels inside the modeled engine it is partly subjected to forces from dynamic pressures resulting from the tangential, radial and axial forces imparted by the flow (see Figure 4-29) – weight and drag also impose forces as will be seen in a later section
    - Velocity along Meanline is meridional, along a meridian (C_m)
    - As will be determined form the baseline geometry derived at Mach 0.475 employing the Meanline Analysis tool, the taper of the Meanline across a component is small and thus the assumption will be made that the axial velocity component of flow and the meridional velocity are the same
    - V_θ, V_r, V_{axial} are the tangential, radial and axial components of particle velocity respectively

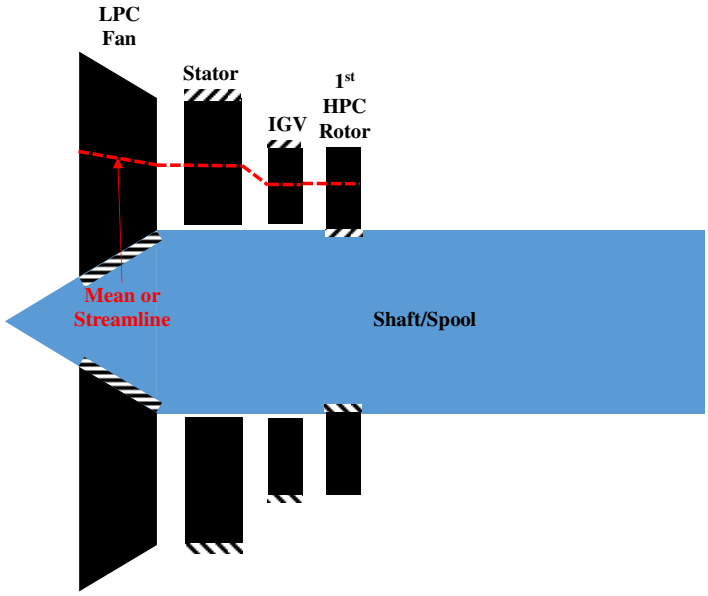


Figure 4.27: Meanline Location

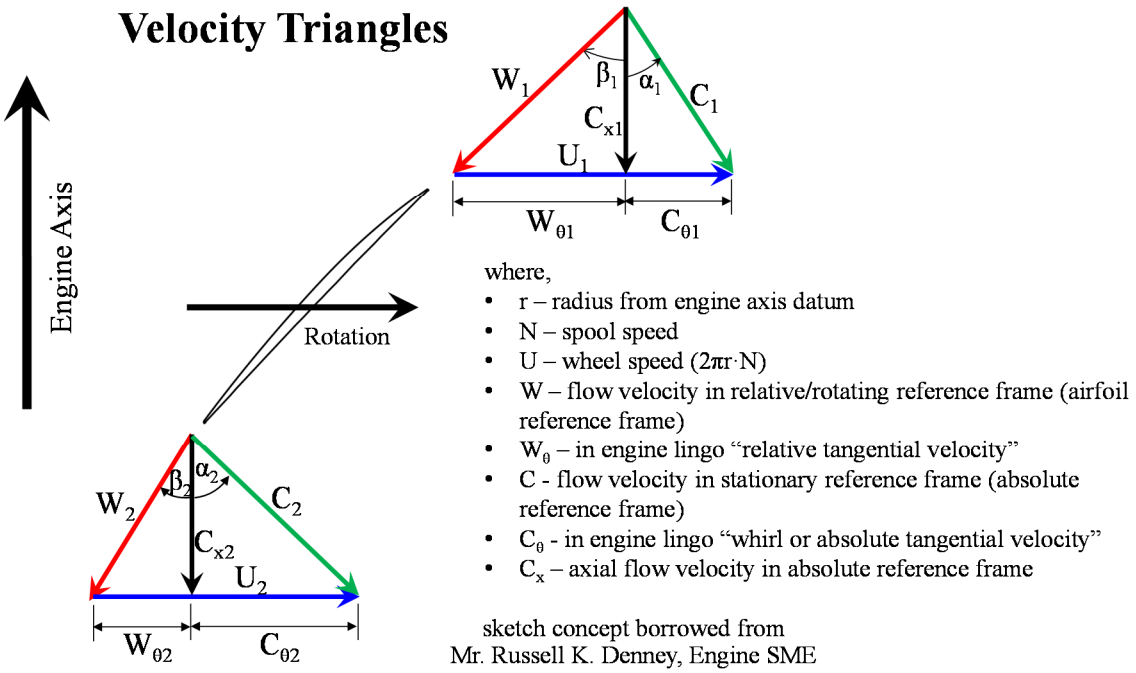


Figure 4.28: Velocity Triangles

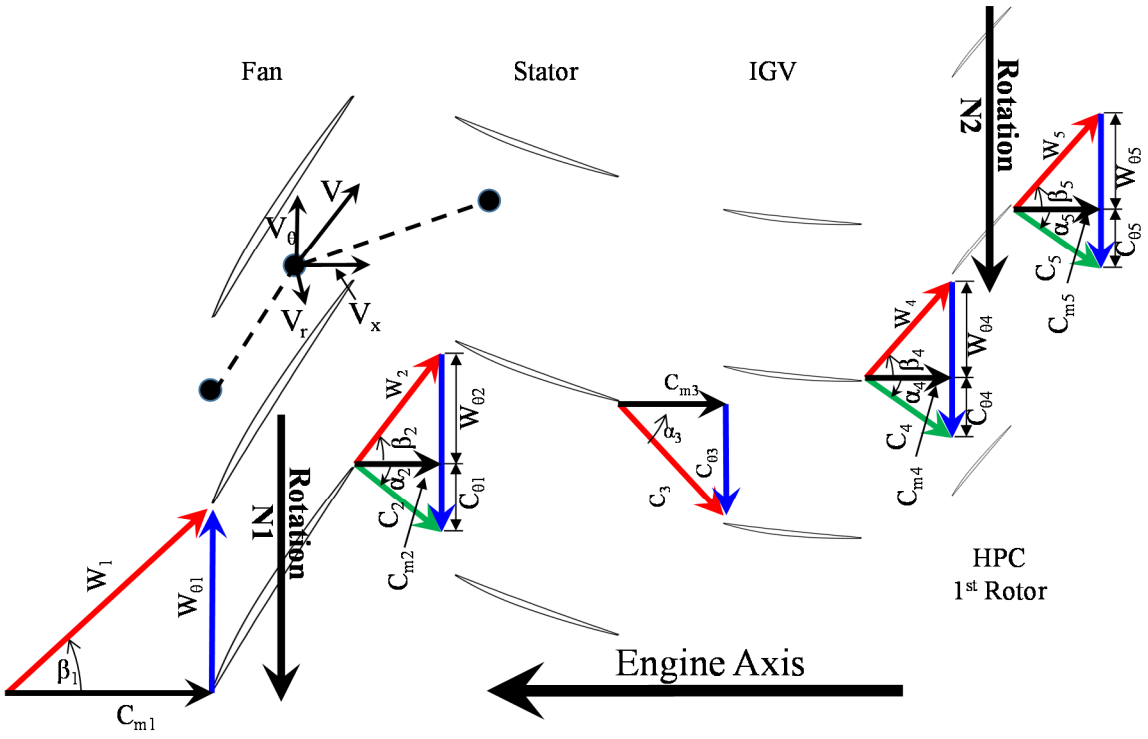


Figure 4.29: FOD Particle Trajectory & Flow Dynamics

- Airfoil & cascade geometry – the NACA 65210 airfoil was chosen for the modeled engine in this thesis because it was employed extensively in legacy axial compressors and ample data is available for it [110, 111] – refer to Figure 4-30 as a guide to match key terminology
    - Airfoil max thickness 10% at 40% chord, max camber 1.1% at 50% chord [112]
    - Solidity, or the ratio of chord to blade pitch σ (c/S) is a key parameter that will be sized by the Meanline Analysis tool at the baseline condition of inlet Mach number of 0.475
    - The stagger angle λ is the radially local incidence of the chord line to the engine axis – as it varies along the radial direction of a rotor it is called twist – this parameter will also be sized by the Meanline Analysis tool

- Remaining parameters will be sized by the Meanline Analysis tool at the baseline inlet Mach number of 0.475 and by methods that will be presented in a later section

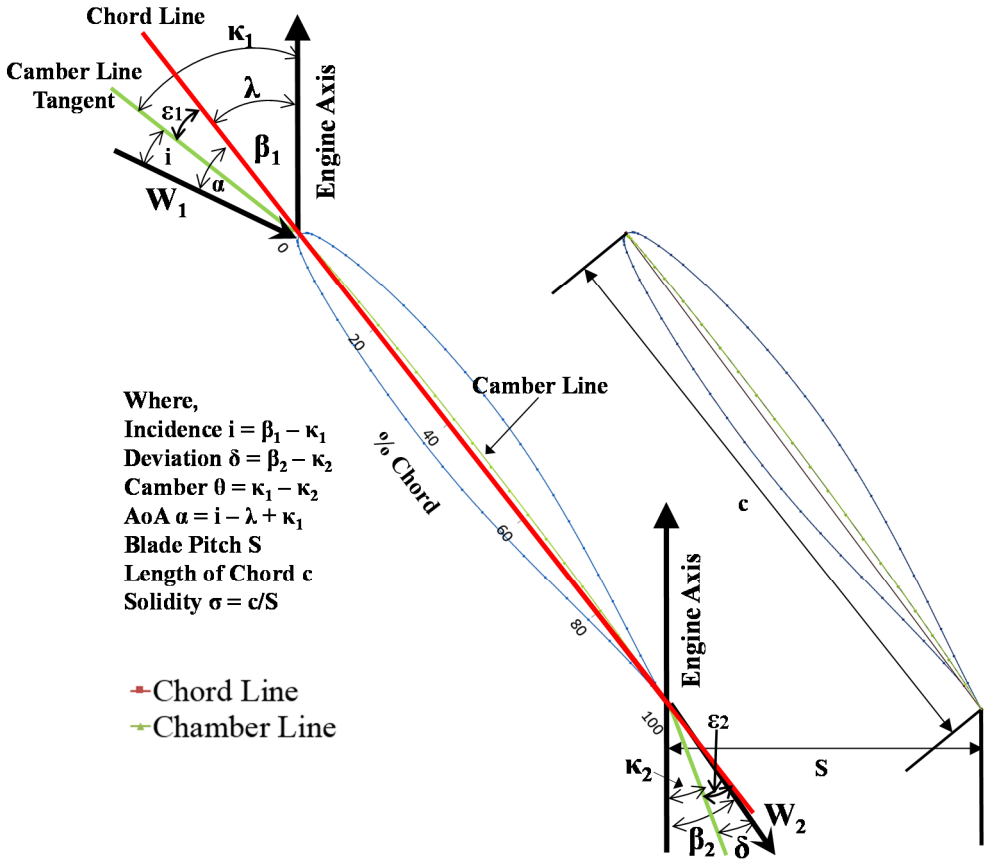


Figure 4.30: Blade and Cascade Geometries

4.7.2 MEANLINE ANALYSIS ASSUMPTIONS

The dynamics and thermodynamics of flow through a jet engine are complex and lengthy, the reader is encouraged to read the cited references for more definitive and insightful overviews of related disciplines [113, 114] - the following assumptions will enable an approximation of the complex nature of flow inside an engine:

- In a cylindrical control volume enclosing a rotor the flow crossing the inlet and outlet is axis-symmetric and steady – leading to small or negligible effect

radially for inlet and outlet components of tangential, radial and axial flow velocity

- Free vortex flow ($r \cdot C_\theta$ = constant) – this is a consequence of no torque acting on a single flow particle about the axial axis – an additional assumption results from free vortex flow: radially along a rotor, two "stations" can be related

$$r_A \cdot C_{\theta A} = r_B \cdot C_{\theta B} \tag{34}$$

$$C_{\theta B} = \frac{r_B}{r_A} \cdot C_{\theta A} \tag{34a}$$

From the velocity triangle,

$$U = C_\theta + W_\theta \tag{35}$$

$$U = U(r) = 2 \cdot \pi \cdot r \cdot N \tag{36}$$

$$W_{\theta B} = U_B(r) - \frac{r_A}{r_B} \cdot (U_A(r) - W_{\theta A}) \tag{37}$$

- Near constancy of axial flow velocity – because axial velocity does not change appreciably as flow travels aft in the compressor section of an engine the density increases and the annulus area of passage decreases to maintain the mass flow – rotors get smaller
- The modeled engine will contain a constant inner shaft or hub diameter – in the engine lingo called a Constant Hub Diameter (CID)
- As the flow in consideration travels in the compressor section of an engine very little heat is added/encountered (<500 °K), the flow will be assumed to behave as a calorifically perfect gas – the gas does not react chemically with internal energy and enthalpy being functions of temperature only and the specific heats are assumed constant

4.7.3 MEANLINE ANALYSIS MODELING PROCEDURE & FORMULAE

By dynamic similarity, the Mach number range of 0.45 to 0.5 for the full scale engine in this thesis will be matched to Murphy's model [96]. Refer to Figure 4-31 to follow the procedure in the context of "Stations". The Meanline analysis tool was developed as a spreadsheet. Static and total flow properties along with geometric parameters are iteratively related via circular references in the spreadsheet – the "baseline" modeled engine geometries are sized iteratively with flow properties at an inlet Mach number of 0.475 (splits the range of inlet Mach numbers to be considered 0.45 to 0.5).

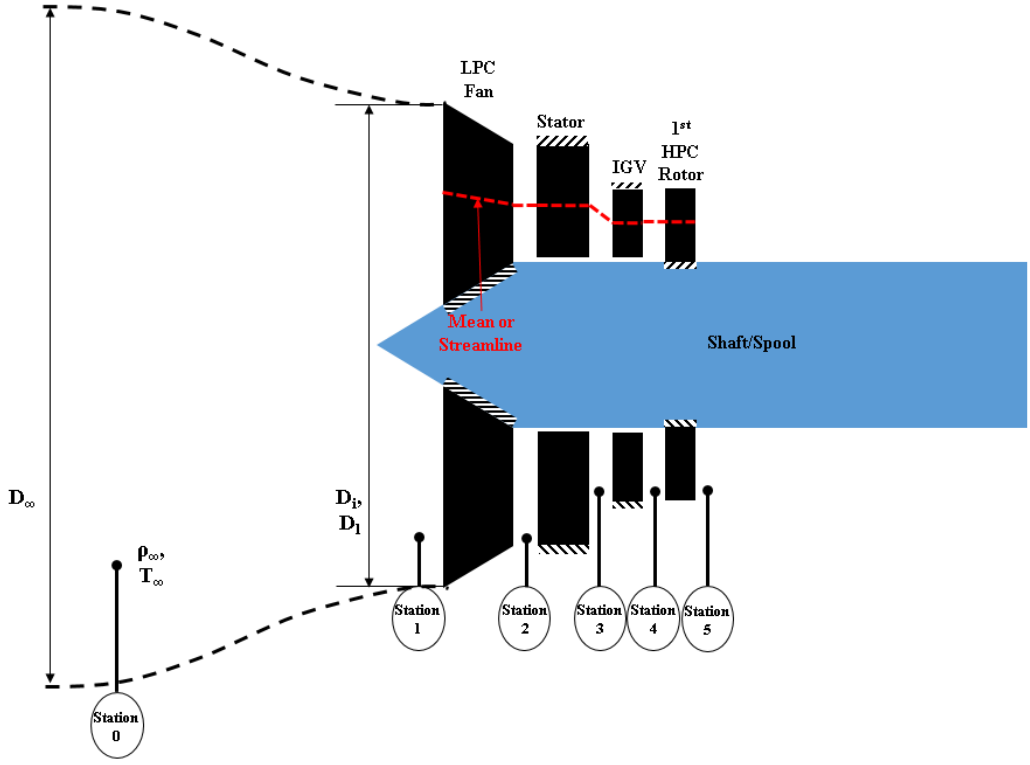


Figure 4.31: Meanline Analysis Engine Stations

Meanline Analysis Basis Discussion, Formulae and Procedure:

- Mass flow is a key parameter of the engine that will be leveraged through-out the formula development and procedure, recall the formulation for mass flow:

$$\dot{m} = U \cdot A \cdot \rho \tag{38}$$

Note: internal to engine U will be replaced by the meridional velocity C_m per reasoning already stated – axial velocity is approximated by meridional velocity

- Flow properties at the exit of a component are the inlet properties of the ensuing component
- The bypass ratio of an F100 engine is 0.36, its overall mass flow is 112.7 kg/s [95, 98] – the core and bypass flow can be estimated by:

$$\dot{m} = \dot{m}_{core} + \dot{m}_{bypass} = 112.7\ kg/s \tag{39}$$

$$BPR = \frac{\dot{m}_{bypass}}{\dot{m}_{core}} = 0.36 \tag{40}$$

Reducing variables through Equation 40, solving for $\dot{m}_{core}$ from Equation 39,

$$\dot{m}_{core} = \frac{\dot{m}}{1+BPR} = \frac{112.7\ kg/s}{1.36} = 82.87\ kg/s \tag{41}$$

$$\dot{m}_{bypass} = \dot{m} \dot{-} m_{core} = 29.83\ kg/s \tag{42}$$

- Three engine parameters are set a priori:
  - Axial Velocity Ratio (AVR) – as the meridional velocity is the axial velocity, its velocity ratio of outlet to inlet is 0.99 (the meridional velocity at the outlet of the fan is 0.99 the meridional velocity at its inlet)
  - Diffusion Factor (DF) - relates peak velocity on Suction Side of airfoil to velocity at trailing edge – Falck suggests that a setting of 0.45 for this parameter is typical in the sizing of compressors, larger values are indicative of stall [105]
  - Blockage Factor (BLK) – a measure of the narrowing of the flow passage as boundary layer growth on the compressor housing increases as the flow travels aft – Falck takes this parameter to be approximately linear from fan inlet to the 5th stage of the compressor in the engine he considers (BLK

ranges from 0.98 to 0.88) [105] – for this thesis his starting value at the fan of 0.98 is kept and by linear interpolation the value of BLK at the HPC rotor is 0.955

- Station 1 – fan inlet
  - At the fan inlet only flow with zero incidence angle with respect to the engine axis is considered in this thesis (no crosswind is assumed) – from the velocity triangle (Figure 4-29) at the fan inlet, with $\alpha_1 = 0$,

$$C_{x1} = C_{m1} = \frac{C_1}{\cos \alpha_1} = C_1 \tag{43}$$

  - Recall the inlet Mach numbers are matched by dynamic similarity to Murphy's engine model at those laboratory test conditions (parameter call out "scaled test conditions") [96] – the full scale engine modeled in this thesis operates at a Standard Day at Sea Level conditions with inlet Mach numbers that range from 0.45 to 0.5 (parameter call out "full scale Sea Level Std. Day") – that match up allows the estimation of the static inlet temperature as follows:

$$M_{i,scaledtestconditions} = M_{i,fullscaleSeaLevelStdDay} \tag{44}$$

$$\frac{U_{i,scaledtestconditions}}{\sqrt{\gamma \cdot R \cdot T_{amb,testconditions}}} = \frac{M_{i,fullscaleSeaLevelStdDay}}{\sqrt{\gamma \cdot R \cdot T_{s1}}} \tag{45}$$

Recall from Equation 38 the definition of mass flow, now including BLK and leveraging Equation 45 and assuming an isentropic process from far-field to fan inlet leads to,

$$\rho_{s1} = \rho_{amb,fullscaleSeaLevelStdDay} \cdot \left(\frac{T_{s1}}{T_{amb,fullscaleSeaLevelStdDay}} \right)^{\frac{1}{\gamma - 1}} \tag{46}$$

$$\dot{m}_{i,fullscaleSeaLevelStdDay} = \dot{m} =$$

$$U_{i,scaledtestconditions} \cdot A_i \cdot \rho_{s1} \cdot BLK_{Fan} \quad (47)$$

$$\dot{m} = M_{i,fullscaleSeaLevelStdDay} \cdot \sqrt{\gamma \cdot R \cdot T_{amb,testconditions}} \cdot \pi \cdot \left\{ {r_{i,tip}}^2 - {r_{i,hub}}^2 \right\} \cdot \rho_{amb,fullscaleSeaLevelStdDay} \cdot \left(\frac{T_{s1}}{T_{amb,fullscaleSeaLevelStdDay}} \right)^{\frac{1}{\gamma-1}} \cdot BLK_{Fan} \quad (47a)$$

Where,

$$A_i = \pi \cdot \left\{ {r_{i,tip}}^2 - {r_{i,hub}}^2 \right\} \quad (48$$

Solving for the static temperature at the fan inlet T_{s1},

$$T_{s1} = [\dot{m}/[M_{i,fullscaleSeaLevelStdDay} \cdot \sqrt{\gamma \cdot R \cdot T_{amb,testconditions}} \cdot \pi \cdot \left\{ {r_{i,tip}}^2 - {r_{i,hub}}^2 \right\} \cdot \rho_{amb,fullscaleSeaLevelStdDay} \cdot BLK_{Fan}]]^{\gamma-1} \cdot BLK \quad (49)$$

For the static pressure (P_{s1}) at the fan inlet,

$$P_{s1} = P_{amb,fullscaleSeaLevelStdDay} \cdot \left(\frac{T_{s1}}{T_{amb,fullscaleSeaLevelStdDay}} \right)^{\frac{\gamma}{\gamma-1}} \quad (50)$$

The stagnation or total properties can now be calculated by means of their thermodynamic relationship to Mach number,

$$\frac{T_{t1}}{T_{s1}} = 1 + \frac{\gamma-1}{2} \cdot {M_{i,fullscaleSeaLevelStdDay}}^2 \quad (51)$$

$$\frac{P_{t1}}{P_{s1}} = \left(1 + \frac{\gamma-1}{2} \cdot {M_{i,fullscaleSeaLevelStdDay}}^2 \right)^{\frac{\gamma}{\gamma-1}} \quad (52)$$

$$\frac{\rho_{t1}}{\rho_{s1}} = \left(1 + \frac{\gamma-1}{2} \cdot {M_{i,fullscaleSeaLevelStdDay}}^2 \right)^{\frac{\gamma}{\gamma-1}} \quad (53)$$

- Entropy, specific heat and static enthalpy are calculated via an ideal properties of air tables [116]:

$$c_{p,1} = f(T_{s1}) \tag{54}$$

$$S_1{}^0 = f(T_{s1}) \tag{55}$$

$$H_{s1} = f(T_{s1})$$

- The meridional velocity at Station 1 (see Figure 4-31) can now be estimated:

$$C_{m1} = \frac{\dot{m}}{\pi \cdot \{r_{i,tip}{}^2 - r_{i,hub}{}^2\} \cdot \rho_{s1} \cdot BLK} \tag{56}$$

- The Meanline analysis Streamline is radially located from the engine axis center – that distance from the engine axis center is the root mean square of the tip and hub radii (see Figure 4-32):

$$r_{rms} = \left\{\frac{r_{tip}{}^2 + r_{hub}{}^2}{2}\right\}^{1/2} \tag{57}$$

For the r_{rms} at the fan inlet (Station 1),

$$r_{rms1} = \left\{\frac{r_{1,tip}{}^2 + r_{1,hub}{}^2}{2}\right\}^{1/2} \tag{57a}$$

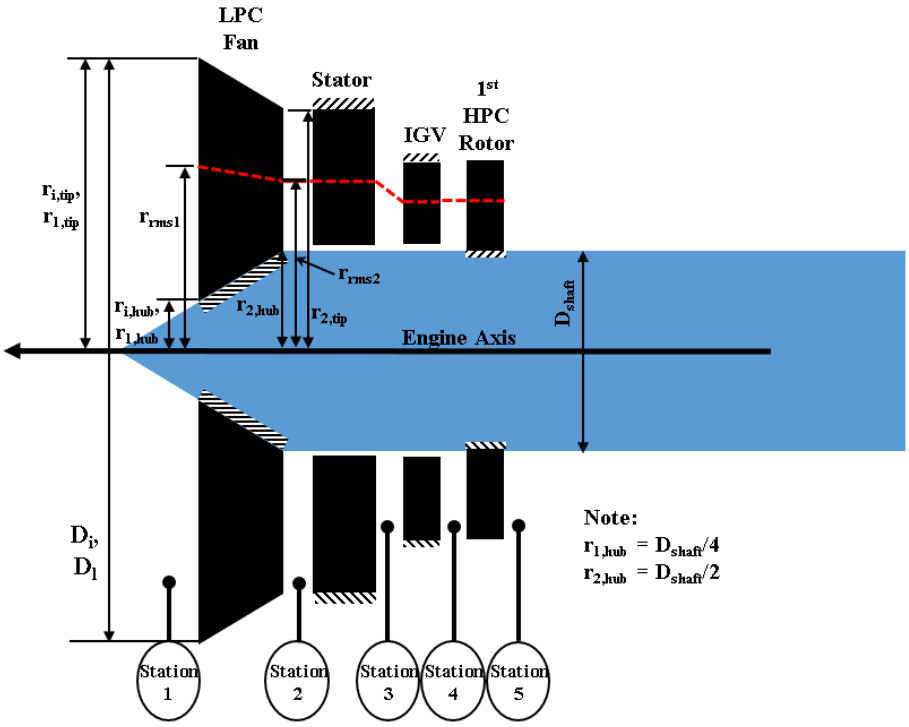


Figure 4.32: Engine Station Geometry

- With the r_{rms} and the spool speeds for the LPC and HPC, N1 and N2 respectively (have units of rpm), at hand the wheel speeds can be calculated:

$$U_1 = U_{rms1} = 2 \cdot \pi \cdot r_{rms1} \cdot \frac{N1}{60} \quad (58)$$

$$U_2 = U_{rms2} = 2 \cdot \pi \cdot r_{rms2} \cdot \frac{N1}{60} \quad (59)$$

$$U_4 = U_{rms4} = 2 \cdot \pi \cdot r_{rms4} \cdot \frac{N2}{60} \quad (60)$$

$$U_5 = U_{rms5} = 2 \cdot \pi \cdot r_{rms5} \cdot \frac{N2}{60} \quad (61)$$

- From the velocity triangle at the fan inlet (see Figure 4-29) and recalling that only flow with zero incidence angle with respect to the engine axis is considered (α_1=0):

$$C_{\theta 1} = C_{m1} \cdot \tan \alpha_1 = 0 \quad (62)$$

$$W_{\theta 1} = U_1 - C_{\theta 1} = U_1 \quad (63)$$

$$\beta_1 = \tan^{-1} \frac{W_{\theta 1}}{C_{m1}} \quad (64)$$

$$W_1 = \frac{C_{m1}}{\cos \beta_1}$$

- Rothalpy, stemming from the concept that the rise of stagnation enthalpy is equal to the specific work done on the fluid (the fan's rotation does work on the flow that passes through it), can now be calculated as follows:

$$Specific\ Work\ on\ Compression = \Delta W_c = U_2 \cdot C_{\theta 2} - U_1 \cdot C_{\theta 1} = H_{t2} - H_{t1} \quad (65)$$

Stagnation or total enthalpy is related to static enthalpy

$$H_t = H_s + 1/2 \cdot C^2 \quad (66)$$

For Station 1 the stagnation enthalpy is H_{t1}

$$H_{t1} = H_{s1} + 1/2 \cdot {C_1}^2 \quad (66a)$$

Equation 65 can be re-arranged with Equations 35 and 66 to reveal a powerful relationship: rothalpy I, across a rotor is constant

$$H_{s1} + {}^1\!/_2 \cdot {W_1}^2 - {}^1\!/_2 \cdot {U_1}^2 = H_{s2} + {}^1\!/_2 \cdot {W_2}^2 - {}^1\!/_2 \cdot {U_2}^2 = I_{1,2} \quad (67)$$

- Relative properties will need to be calculated to estimate pressure losses and entropy rises across the fan:

$$M_{W_1} = \frac{W_1}{\sqrt{\gamma \cdot R \cdot T_{s1}}} \quad (68)$$

$$P_{t1,W_1} = P_{s1} \cdot \left(1 + \frac{\gamma - 1}{2} \cdot {M_{W_1}}^2\right)^{\frac{\gamma}{\gamma - 1}} \quad (69)$$

$$T_{t1,W_1} = \frac{H_{s1} + 1/2 \cdot {W_1}^2}{c_{p,1}} \quad (70)$$

- As the flow travels from Station 1 to 2 (across the fan) it's design total pressure ratio can be used along with a fan stage performance map (see Figure 4-33) to estimate the actual total pressure ratio affected by the operating conditions and pressure losses as follows [108]:
  - With an estimated design total fan Pressure Ratio (FPR) of 3.8 for three stages, the design total pressure ratio for a single fan stage modeled in this thesis is approximated by:

$$\left({}^{P_{t2}}\!/_{P_{t1}}\right)_{Design} \approx 3.8^{1/3} = 1.56 \quad (71)$$

  - Total pressure losses across the fan are estimated with profile and endwall losses, ω_p and ω_e, following Falck's methodology [105]:

$$\Delta P_{2-1} = \left(\omega_p + \omega_e\right) \cdot \left(P_{t1,W_1} - P_{s1}\right) \quad (72)$$

  - Correcting for operating conditions and losses:

$$\left(P_{t2}/P_{t1}\right)_{Corrected} =$$

$$\left(P_{t2}/P_{t1}\right)_{Design} \cdot (\%\ Pressure\ Ratio)_{PerformanceMap} - \frac{\Delta P_{2-1}}{P_{t2}}$$

(73)

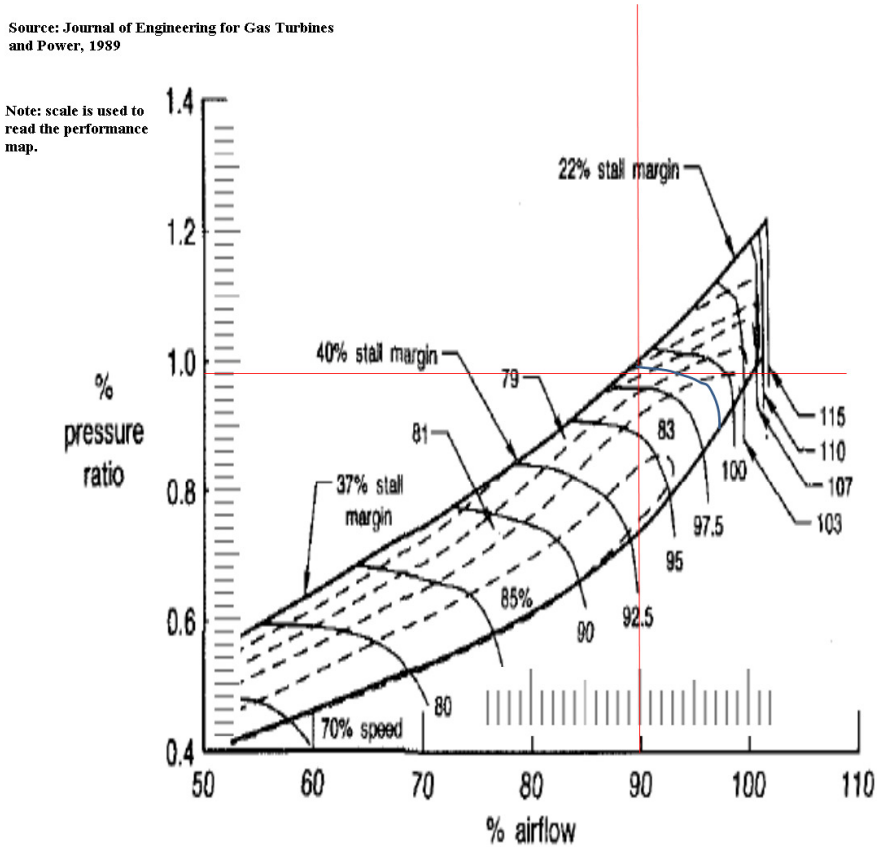


Fig. 8 F100-PW-229 fan map

Figure 4.33: F100-PW-229 Fan Performance Map

- To extract the ratio of total pressure ratios from the performance map (reported as % pressure ratio on map) and the isentropic efficiency of the fan stage (reported as plateaus of constant % efficiency) the corrected mass flow ratio (reported as % airflow on the map) and the corrected N1speed ratio (reported as % curves of constant N1 ratio

magnitudes on the map, the % sign is omitted in map) must be estimated from operating conditions as follows:

$$Total\ Temperature\ Ratio = \theta = \frac{T_{t1}}{T_{amb,fullscaleSeaLevelStdDay}} \quad (74)$$

$$Total\ Pressure\ Ratio = \delta = \frac{P_{t1}}{P_{amb,fullscaleSeaLevelStdDay}} \quad (75)$$

Assuming operating condition and design ratios of mass flows and N1 speed are similar in magnitude at 100% power setting,

$$\text{Corrected Mass Flow Ratio} = \left(\frac{\dot{m}}{\dot{m}_{Design}}\right)_{Corrected} \approx \frac{\dot{m}}{\dot{m}_{Design}} \cdot \frac{\sqrt{\theta}}{\delta} = \frac{\sqrt{\theta}}{\delta} \quad (76)$$

Likewise the Corrected N1 speed ratio becomes,

$$\text{Corrected N1 Speed Ratio} = \left(\frac{U_1}{U_{1,Design}}\right)_{Corrected} \approx \frac{U_1}{U_{1,Design}} \cdot \frac{1}{\sqrt{\theta}} = \frac{1}{\sqrt{\theta}} \quad (76)$$

- To estimate the efficiency of only the fan, instead of employing the fan stage isentropic efficiency $\eta_{isentropic}$ extracted from the performance map, polytropic efficiency η_p will be used instead – polytropic efficiency is the result of "chopping" up the compression process into individual steps that aggregately approximate the isentropic efficiency

$$\eta_p = \frac{\gamma-1}{\gamma} \cdot \frac{\log\left(P_{t2}/P_{t1}\right)}{\log\left(T_{t2}/T_{t1}\right)} \quad (77)$$

Where,

$$\frac{T_{t2}}{T_{t1}} = 1 + \frac{\left\{\left(P_{t2}/P_{t1}\right)^{\frac{\gamma-1}{\gamma}} - 1\right\}}{\eta isentropic} \quad (78)$$

- Station 2 – fan outlet, stator inlet

  With Station 1 static and total properties, Station 2 calculations can proceed as follows:

  - $C_{m2} = C_{m1} \cdot AVR_{Fan}$ (79)
  - $r_{rms2} = \left\{\frac{r_{2,tip}{}^2 + r_{2,hub}{}^2}{2}\right\}^{1/2}$ (80)
  - U_2 was presented in Equation 59
  - The Stage Load Coefficient (φ) correlates stage efficiency, it is a parameter whose first guess (it will be iterated in the tool) is used to estimate the whirl velocity in Station 2 as follows:

    $C_{\theta 2} = \varphi \cdot U_2 + \frac{r_{rms1}}{r_{rms2}} \cdot C_{\theta 1}$ (81)
  - From the velocity triangle at Station 2 (see Figure 4-29):

    $W_{\theta 2} = U_2 - C_{\theta 2}$ (82)

    $C_2 = \left(C_{m2}{}^2 + C_{\theta 2}{}^2\right)^{1/2}$ (83)

    $W_2 = \left(C_{m2}{}^2 + W_{\theta 2}{}^2\right)^{1/2}$ (84)

    At this point φ is recalculated (it is linked to Equation 81 via a circular reference) by,

    $\varphi = \frac{C_{\theta 2} \cdot r_{rms2} - C_{\theta 1} \cdot r_{rms1}}{U_2 \cdot r_{rms2}}$ (85)

    $\alpha_2 = \tan^{-1}\left(\frac{C_{\theta 2}}{C_{m2}}\right)$ (86)

    $\beta_2 = \tan^{-1}\left(\frac{W_{\theta 2}}{C_{m2}}\right)$ (87)
  - Recall that rothalpy is constant across Station 1 and 2, leading to:

    $H_{s2} = I_{1,2} - \frac{W_2{}^2}{2} + \frac{U_2{}^2}{2}$ (88)

From H_{s2} $c_{p,2}$ and T_{s2} are extracted from ideal properties of air tables (embedded in Meanline Analysis tool),

$$c_{p,2} = c_{p,2}(H_{s2}) \tag{89}$$

$$T_{s2} = T_{s2}(H_{s2}) \tag{90}$$

$$\rho_{s2} = \frac{\dot{m}}{\pi \cdot \{r_{2,tip}{}^2 - r_{2,hub}{}^2\} \cdot C_{m2} \cdot BLK_{Stator}} \tag{91}$$

From the ideal gas law or by applying Equation 50 for Station 2 (they are thermodynamically equivalent),

$$P_{s2} = \rho_{s2} \cdot R \cdot T_{s2} \tag{92}$$

Entropy is extracted from ideal properties of air tables (embedded in Meanline Analysis tool) and entropy increase due to losses calculated by Falck's procedure [105],

$$S_2{}^0 = S_2{}^0(H_{s2}) + (S_2{}^0 - S_1{}^0)_{losses} \tag{93}$$

Assuming specific heats are nearly constant ($c_{p,1} \approx c_{p,2}$), Equation 78 can be used to estimate total enthalpy,

$$\frac{H_{t2}}{H_{t1}} = \frac{c_{p,2} \cdot T_{t2}}{c_{p,1} \cdot T_{t1}} = \left[1 + \frac{\left\{\left({}^{P_{t2}}/_{P_{t2}}\right)^{\frac{\gamma-1}{\gamma}} - 1\right\}}{\eta_{isentropic}}\right] \tag{94}$$

$$M_{C_2} = \frac{C_2}{\sqrt{\gamma \cdot R \cdot T_{s2}}} \tag{95}$$

$$\rho_{t2} = \rho_{s2} \cdot \left(1 + \frac{\gamma-1}{2} \cdot M_{C_2}{}^2\right)^{\frac{1}{\gamma-1}} \tag{96}$$

- At this point, the Meanline Analysis tool can be employed to "size" the fan

- The stagger angle of the fan blade with respect to the engine axis center can be estimated with a NASA procedure [115] – from the stagger angle other airfoil angles will be calculated

  With the Stage Loading Coefficient,

$$Axial\ Solidity = \sigma_x = \frac{2 \cdot \cos\beta_2}{\varphi \cdot \cos\beta_1} \cdot \sin(\beta_1 - \beta_2) \quad (97)$$

$$a = \frac{\sigma_x - \sin\beta_1 \cdot \cos\beta_1 + \sin\beta_2 \cdot \cos\beta_2}{\sin\beta_1 - \sin\beta_2} \quad (98)$$

$$c_\theta = \sin{\beta_1}^2 - a \cdot (\cos\beta_1 - \cos\beta_2) - \sin{\beta_2}^2 \quad (99)$$

$$Fan\ Stagger\ Angle = \ \lambda_{Fan} = \tan^{-1}\frac{c_\theta}{\sigma_x} \quad (100)$$

  blade metal angles are estimated with the aid of stagger angle λ_{Fan} and the angles between the camber line tangents at the leading and trailing edge and the chord line, ε_1 and ε_2 respectively (see Figure 4-30 for full list of blade angles calculated),

$$\kappa_1 = \lambda_{Fan} + \varepsilon_1 \quad (101)$$

$$\kappa_2 = \lambda_{Fan} - \varepsilon_2 \quad (102)$$

$$i = \beta_1 - \kappa_1 \quad (103)$$

$$\alpha = \kappa_1 + i - \lambda_{Fan} \quad (104)$$

$$\delta = \beta_2 - \kappa_2 \quad (105)$$

  The blade camber is calculated by,

$$\theta = \kappa_2 - \kappa_1 \quad (106)$$

  At any point radially on a rotor the twist is linearly approximated by,

$$\lambda(r) = \left(\lambda_{tip} - \lambda_{hub}\right) \cdot \frac{(r - r_{rms})}{r_{tip} - r_{hub}} + \lambda_{rms} \quad (107)$$

- The height or span of the fan rotor can now be sized via iteration – recall that Meanline Analysis tool links geometric parameters to static and total properties via circular references (the tool is in the form of a spreadsheet)

  From mass flow at Station 2,

$$D_{FanOutlet} = 2 \cdot r_{2,tip} = 2 \cdot \sqrt{\frac{\dot{m}}{\pi \cdot \rho_{s2} \cdot C_{m2} \cdot BLK_{Fan}} + r_{2,hub}{}^2} \quad (108)$$

- The Meanline Analysis tool can now size the average r_{rms} chord of the airfoil

  With known number of fan blades, the pitch at r_{rms} (at inlet and outlet of fan rotor) can be calculated,

$$Fan\ Blade\ Pitch = S = \frac{2 \cdot \pi \cdot r_{rms}}{Fan\ No.Blades} \quad (109)$$

  The inverse of cascade solidity $1/\sigma$ (S/c) is calculated with the Hearsey Method [105] employing relative flow angles, total losses ($\omega = \omega_e + \omega_p$, already calculated for pressure losses and entropy rise) and a fixed Diffusion Factor with a typical value for compressor sizing of 0.45

$$\frac{1}{\sigma_{Fan}} = 0.004 e^{6.1677 \cdot DF^{1.436794}} \cdot \left(\frac{\cos \beta_1}{\cos \beta_2}\right)^2 \cdot \frac{2}{\omega \cdot \cos \beta_2} \quad (110)$$

  the average r_{rms} chord is estimated by,

$$C_{Fan,rms} = \frac{\sigma_{Fan} \cdot (S_1 + S_2)}{2} \quad (111)$$

- Station 3 – stator outlet, IGV inlet
  - $C_{m3} = C_{m2} \cdot AVR_{Stator}$ (112)
  - $C_{\theta 3} = C_{m3} \cdot \tan \alpha_3$ (113)

Assuming that $\alpha_3 = \lambda_{Stator}$ because stators have no twist, only a constant stagger

- λ_{Stator} is calculated with the same procedure as λ_{Fan}
- $C_3 = \frac{C_{\theta 3}}{\cos \alpha_3}$ (114)
- Recall that at the stator no work is done on the flow, thus the stagnation properties at the inlet of the IGV remain the same as those at the outlet of the stator

$$H_{t3} = H_{t2} \quad (115)$$

$$H_{s3} = H_{t2} - \frac{{C_3}^2}{2} \quad (116)$$

The flow is now entering the core,

$$\dot{m}_{core} = C_{m3} \cdot A_{inlet,IGV} \cdot \rho_{s3} \cdot BLK_{IGV} \quad =$$

$$C_{m3} \cdot A_{inlet,IGV} \cdot \rho_{s2} \cdot \left(\frac{T_{s3}}{T_{s2}}\right)^{\frac{1}{\gamma-1}} \cdot BLK_{IGV} \quad (117)$$

Solving for T_{s3},

$$T_{s3} = \left(\frac{\dot{m}_{core}}{C_{m3} \cdot \pi \cdot \left\{ {r_{Stator,tip}}^2 - \frac{{D_{shaft}}^2}{2} \right\} \cdot \rho_{s2} \cdot BLK_{IGV}} \right)^{\gamma-1} \cdot T_{s2} \quad (118)$$

$$\rho_{s3} = \rho_{s2} \cdot \left(\frac{T_{s3}}{T_{s2}}\right)^{\frac{1}{\gamma-1}} \quad (119)$$

$$P_{s3} = P_{s2} \cdot \left(\frac{T_{s3}}{T_{s2}}\right)^{\frac{\gamma}{\gamma-1}} \quad (119)$$

$$c_{p,3} = H_{s3}/T_{s3} \quad (120)$$

$$M_{C_3} = \frac{C_3}{\sqrt{\gamma \cdot R \cdot T_{s3}}} \quad (121)$$

For total properties,

$$\frac{T_{t3}}{T_{s3}} = 1 + \frac{\gamma-1}{2} \cdot {M_{C_3}}^2 \quad (122)$$

$$\frac{P_{t3}}{P_{s3}} = \left(1 + \frac{\gamma-1}{2} \cdot {M_{C_3}}^2\right)^{\frac{\gamma}{\gamma-1}} \quad (123)$$

$$\frac{\rho_{t3}}{\rho_{s3}} = \left(1 + \frac{\gamma-1}{2} \cdot {M_{C_3}}^2\right)^{\frac{\gamma}{\gamma-1}} \quad (124)$$

Rrms through the stator remains constant,

$$r_{rms,Stator} = \left\{\frac{{r_{Stator,tip}}^2 + \frac{{D_{shaft}}^2}{2}}{2}\right\}^{1/2} \quad (125)$$

- The stator is assumed to have the same number of blades as the fan, thus the procedure to calculate blade pitch S and cascade solidity σ is the same as that of the fan

$$c_{Stator,rms} = \frac{\sigma_{Stator} \cdot (S_2 + S_3)}{2} \quad (126)$$

- The diameter of the stator at inlet and outlet are sized just as the Fan

$$D_{StatorInlet} = 2 \cdot r_{StatorInlet,tip} = 2 \cdot \sqrt{\frac{\dot{m}_{core}}{\pi \cdot \rho_{s2} \cdot C_{m2} \cdot BLK_{Stator}} + \frac{{D_{shaft}}^2}{2}} \quad (127)$$

$$D_{StatorOutlet} = 2 \cdot r_{StatorOutlet,tip} = 2 \cdot \sqrt{\frac{\dot{m}_{core}}{\pi \cdot \rho_{s3} \cdot C_{m3} \cdot BLK_{Stator}} + \frac{{D_{shaft}}^2}{2}} \quad (128)$$

- Station 4 – IGV outlet, 1st HPC rotor inlet

All calculations are performed per the procedure for Station 1 (fan inlet) with the exception that is not zero and the 1st HPC rotor employs the performance map for the compression section of an F100 engine (see Figure 4-34) [108]. An approximation was made to estimate the single fan FPR, it will be assumed that corrected, design total pressure ratio for the 1st HPC rotor blade stage is approximately:

$$\left(\frac{P_{t5}}{P_{t4}}\right)_{Corrected} = \left[\frac{OPR}{\left(P_{t2}/P_{t1}\right)_{Design}}\right]^{\frac{1}{13}} \cdot (\%Pressure\ Ratio)_{PerformanceMap} \quad (129)$$

Note that the performance map for the compressor specifically calls out Sea Level takeoff (SLTO on plot) – this is the very condition that this thesis focuses on, thus the corrected total pressure ratio across the rotor can be calculated

$$\left(\frac{P_{t5}}{P_{t4}}\right)_{Corrected} = 1.262 \cdot 0.987 = 1.245594 \quad (129a)$$

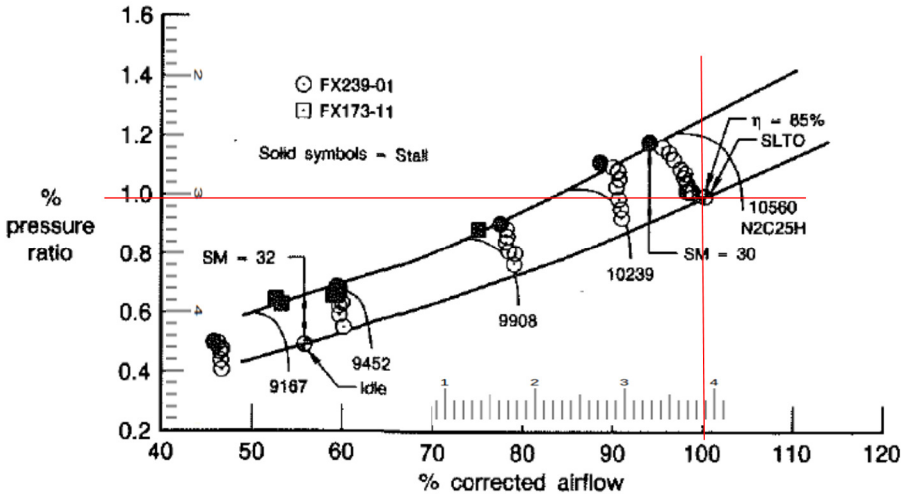


Fig. 10 F100-PW-229 compressor map

Figure 4.34: F100-PW-229 Compressor Map

- Station 5 – 1st HPC rotor outlet

  All calculations are performed per the procedure for Station 2 (fan outlet).

4.7.4 Meanline Analysis Simulation

The Meanline Analysis tool was ran 10,000 times – randomly, 10,000 inlet Mach numbers were generated and 10,000 values for all properties already detailed in Section 4.7.3 were calculated using Microsoft Excel's Data Table feature [117]. From the 10,000

data point sets distributions were developed with SAS Statistical Discovery Software JMP® - Beta distributions were developed that simulated static properties and flow velocities needed for the particle aspiration and kinetics models of Sections 4.6 and 4.8 respectively – the distributions are presented in Appendix D along with the code for the simulation.

4.8 IN-ENGINE PARTICLE KINETICS MODEL

This thesis focuses on small, Hard Body FOD – specifically small, metallic, debris particles on the runway during the early takeoff phase. Though small (most of the case study work in a later section will focus on steel particles 1.33 mm and 3.2 mm in diameter) the particles are dense, thus we must speak of kinetics and not kinematics. Kinematics focuses on the displacements of particles whose masses are negligible or not accounted for, while this thesis focuses on very real damage caused by small, Hard Body FOD – the mass/size of the debris particles and the forces that propel then from ingestions through the fan face up to impact at a blade on the 1st HPC rotor are the stuff or kinetics.

4.8.1 EQUATIONS OF MOTION

4.8.1.1 Plane, Polar Coordinate Kinetics

To develop the equations of motion for a particle that is ingested into the engine (beyond the fan face the regime is ingestion, from ground up to fan face aspiration) it is assumed that the motion of the particle in the polar coordinate system, whose datum is the engine centerline axis, is decoupled from the axial motion. The assumption is supported from the previous assumption that a cylindrical control volume across any of the engine components is axis-symmetric – this assumption leads to flow properties that are radially independent of axial behavior – e.g. the meridional velocity of the flow, which represents the axial velocity, is calculated across component Stations by means of an Average

Velocity Ratio (AVR) while tangential velocities are must add up to the local wheel speed (see Figure 4-28 and go to Section 4.7).

A coordinate system in the engine is not an inertial reference frame – there is rotation. To determine the force system that governs the trajectory of a particle in a reference frame rotating with an angular velocity ω, the relationship of an inertial reference frame collocated at that engine axis center and the rotating reference frame must be understood. It is assumed that the datum for a fixed Cartesian coordinate system is the engine centerline axis – the rotating reference frame shares this same datum. If an arbitrary vector n is rigidly fixed to the rotating reference frame, then the time derivative of that vector about the fixed or inertial reference frame is:

$$\frac{d\vec{n}}{dt} = \vec{\omega} \times \vec{n} \tag{130}$$

If the radial vector extending from the datum is decomposed into its components on the rotating frame (Equation 131) and the time derivative is taken, the first parenthesis in Equation 132 reveals that the unit vectors remains fixed, while the second parenthesis represents the time derivative of $\vec{r}$ on the rotating frame. After re-arranging the terms in the first parenthesis in Equation 132 (see Equation 132a), Equation 133 reveals the relationship between the inertial and rotating reference frames.

$$\vec{r} = r_x\vec{x} + r_y\vec{y} + r_z\vec{z} \tag{131}$$

$$\frac{d\vec{r}}{dt} = \left(r_x\frac{d\vec{x}}{dt} + r_y\frac{d\vec{y}}{dt} + r_z\frac{d\vec{z}}{dt}\right) + \left(\frac{dr_x}{dt}\vec{x} + \frac{dr_y}{dt}\vec{y} + \frac{dr_z}{dt}\vec{z}\right) \tag{132}$$

$$r_x\frac{d\vec{x}}{dt} + r_y\frac{d\vec{y}}{dt} + r_z\frac{d\vec{z}}{dt} = r_x(\vec{\omega} \times \vec{x}) + r_y(\vec{\omega} \times \vec{y}) + r_z(\vec{\omega} \times \vec{z}) =$$

$$\vec{\omega} \times \left(r_x\vec{x} + r_y\vec{y} + r_z\vec{z}\right) = \vec{\omega} \times \vec{r} \tag{132a}$$

$$\left(\frac{d\vec{r}}{dt}\right)_{InertialEngineCenterlineAxis} = \left(\frac{d\vec{r}}{dt}\right)_{RotatingEngineReferenceFrame} + \vec{\omega} \times \vec{r} \tag{133}$$

Considering Newton's second law, assuming that a particle that is ingested keeps it's mass and taking another time derivative (obeying the chain rule) reveals that external forces on a particle must match the forces from both the inertial and rotating reference frame see Equation 134. In reference to Equation 134: after the cross product is taken in the second term it shares the same direction as the first term, those two terms are lumped as "centrifugal" forces - they account for forces that pull the particle radially, the third term is the Euler that force accounts for forces on the particle perpendicular to the radial vector and the fourth term is the Coriolis force that is always perpendicular to velocity vector.

$$\vec{F} = \left(m\frac{d^2\vec{r}}{dt^2}\right)_{RotatingEngineReferenceFrame} + m\vec{\omega} \times (\vec{\omega} \times \vec{r}) + m\left(\frac{d\vec{\omega}}{dt}\right)_{RotatingEngineReferenceFrame} \times \vec{r} + 2m\vec{\omega} \times \left(\frac{d\vec{r}}{dt}\right)_{RotatingEngineReferenceFrame} \quad (134)$$

The time derivative of angle θ at any point and the angular velocity are related by Equation 135 - said relationship evinces another in Equation 136 that will be employed to separate Equation 134 in terms of force magnitudes instead of vectors. Figure 4-35 presents the particle inside the engine in the context of the datum, the engine centerline axis – the figure presents the particle force system that matches the magnitudes of the Euler, Coriolis and "Centrifugal" forces.

$$\dot{\theta}(t) = \omega(t) \quad (135)$$

$$\ddot{\theta}(t) = \dot{\omega}(t) \quad (136)$$

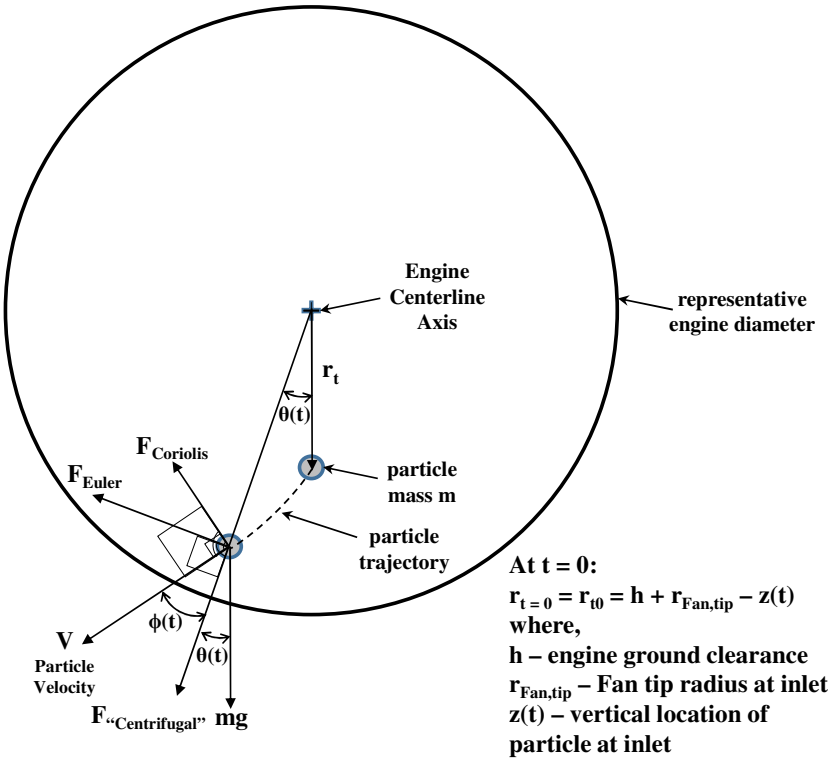


Figure 4.35: In-Engine Particle Polar Force/Displace System

The assumptions of axis-symmetry due to the choice of a cylindrical control volume enables the key assumptions for the polar particle kinetics:

- "Centrifugal" forces on the particle are negligibly affected by radial engine flow as radially there is little flow in an axial compressor – those forces are affected by particle weight and drag similar to that experience by traveling through stagnant flow
- Euler forces on the particle are most affected by flow dynamics – they encounter the tangential velocities across component Stations
- Coriolis forces on the particle are affected by a component of Euler forces along the trajectory tangent

With the assumptions on the nature of forces on the particle, the relationships revealed in Equations 137 through 139a for the angle between the radial vector and particle

velocity (see Figure 4-35) the polar force/displacement system is described with any two force sets (any two of "Centrifugal", Coriolis or Euler force sets). Choosing "Centrifugal" and Euler forces, the summation of forces is presented by Equations 140 and 141.

$$V(t) \cdot \cos\phi(t) = \dot{r}(t) \tag{137}$$

$$V(t) \cdot \sin\phi(t) = r(t) \cdot \dot{\theta}(t) \tag{138}$$

$$\cos\phi(t)^2 + \sin\phi(t)^2 = 1 = \left(\frac{\dot{r}(t)}{V(t)}\right)^2 + \left(\frac{r(t)\cdot\dot{\theta}(t)}{V(t)}\right)^2 \tag{139}$$

$$V(t) = \sqrt{(\dot{r}(t))^2 + \left(r(t) \cdot \dot{\theta}(t)\right)^2} \tag{139a}$$

$\sum F_{Centrifugal}$:

$$m\left(r(t) \cdot \dot{\theta}(t)^2 + \ddot{r}(t)\right) = \frac{1}{2} \cdot \rho_{Station} \cdot V^2 \cdot A_{FOD} \cdot (1 - C_D) \cdot \cos\phi(t) + mg \cdot \cos\theta(t) \tag{140}$$

Substituting for $\cos\phi(t)$ from Equation 137 and V(t) from Equation 139a,

$$m\left(r(t) \cdot \dot{\theta}(t)^2 + \ddot{r}(t)\right) = \frac{1}{2} \cdot \rho_{Station} \cdot \sqrt{\left(\dot{r}(t)\right)^2 + \left(r(t) \cdot \dot{\theta}(t)\right)^2} \cdot A_{FOD} \cdot (1 - C_D) \cdot \dot{r}(t) + mg \cdot \cos\theta(t) \tag{140a}$$

$\sum F_{Euler}$:

$$m \cdot r(t) \cdot \ddot{\theta} = \frac{1}{2} \cdot \rho_{Station} \cdot V_{TangentialFlowVelocity}{}^2 \cdot A_{FOD} \cdot (1 - C_D) - mg \cdot \sin\theta(t) \tag{141}$$

Initial conditions to solve the coupled differential equation system are as follows (recall for this thesis we consider incoming particle and flow to have no incidence angle with respect to the engine centerline axis):

$\theta(t = 0) = 0$

$r(t = 0) = {}^{D_l}/_2 + h + z$

$\dot{\theta}(t = 0) = 0$

$\dot{r}(t=0)=0$

Where,

$z - z_{corrected}$ from Equation 23 for corrected vertical location of particle at inlet

h – engine ground clearance or ground to engine highlight clearance

D_l – engine inlet diameter

4.8.1.2 Axial Kinetics

The axial kinetics of the FOD particle as it travels inside the engine are affected by the meridional velocities calculated with Meanline Analysis tool. Equation presents the model for the axial force/displacement system that has as an inertial reference frame the fan face.

$$m \cdot \frac{d^2X}{dt^2} = \frac{1}{2} \cdot \rho_{Station} \cdot {C_m}^2 \cdot A_{FOD} \cdot (1 - C_D) \quad (142)$$

Initial conditions to solve the differential equation are as follows:

$X(t=0)=0$

$\dot{X}(t=0) = V_{x,inlet} = \left(\frac{dx}{dt}\right)_{inlet}$

Where,

$V_{X,inlet}$ or $\left(\frac{dx}{dt}\right)_{inlet}$ - are the inlet velocities of Section 4.6.2 (Aspiration Model formulation, Equation 32)

A discussion is required to specify per component Station what tangential and axial velocities apply and how they are scaled radially (an assumption made in Section 4.7) and axially (the Meanline Analysis tool calculates the tangential velocities at each Station, the differences must also be accounted for) respectively. The modeling per component Station of the tangential velocities on the FOD particle is as follows (see Figure 4-36):

- Station 1 – the particle is ingested with no incidence angle with respect to the engine centerline axis, it is subjected to $W_{\theta 1}$ (equal to the wheel speed at this Station) and C_{m1} –as no documented assumption could be made on how the static density of the flow changes as it travels through a component, the assumption is that it stays constant, thus for this Station the density is ρ_{s1}
- Station 2 – at the fan outlet the particle is subjected to $W_{\theta 2}$ and C_{m2}, at the stator inlet the tangential velocity become $C_{\theta 2\ because}$ that component is fixed while C_{m2} is kept - for this Station the flow density is ρ_{s2}
- Station 3 - at the stator outlet the particle is subjected to $C_{\theta 3}$ and C_{m3}, this combination remains at the IGV inlet - for this Station the flow density is ρ_{s3}
- Station 4 - at the IGV outlet the particle is subjected to $C_{\theta 4}$ and C_{m4}, while at the 1st HPC rotor inlet the particle is subjected to $W_{\theta 4}$ and C_{m4} - for this Station the flow density is ρ_{s4}
- Station 5 - at the 1st HPC rotor outlet the particle is subjected to $W_{\theta 5}$ and C_{m5} - for this Station the flow density is ρ_{s5}

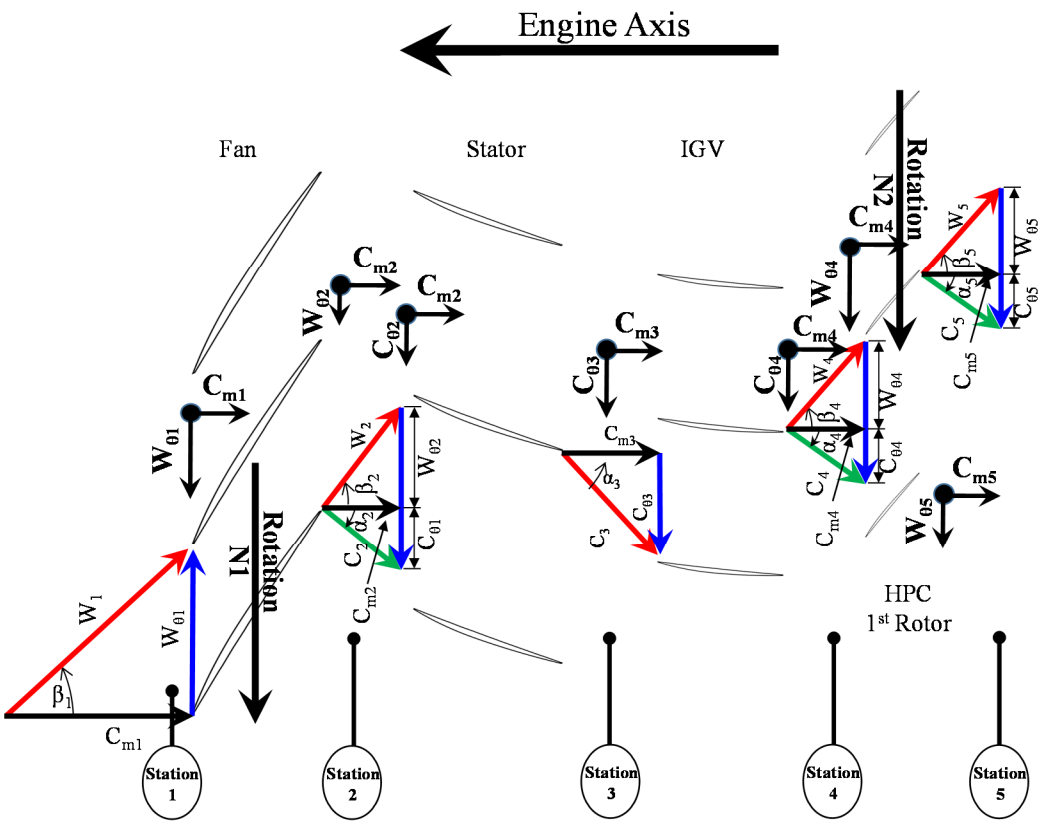


Figure 4.36: Flow Velocities Acting on FOD Particles

The tangential velocities were linearly interpolated across a component Station – the linear interpolations are built into the In-Engine Particle Kinetics simulation code developed with Wolfram Mathematica® 10 presented in Appendix E. The following detail how that interpolation is made:

- Recall the Free Vortex condition of $r \cdot C_\theta$ = constant

$$r_{rms} \cdot C_{\theta,rms} = r(t) \cdot C_\theta \tag{143}$$

$$C_\theta = \frac{r_{rms}}{r(t)} \cdot C_{\theta,rms} \tag{144}$$

- Recall that there is a relationship between whirl velocity and wheel speed C_θ = U - W_θ (where U = r(t) ·ω, ω is the angular velocity in rads/s)

$$r_{rms} \cdot \left(r_{rms} \cdot \omega - W_{\theta,rms}\right) = r(t) \cdot \left(r(t) \cdot \omega - W_\theta\right) \tag{145}$$

Solving for W_θ,

$$W_\theta(r) = \frac{r(t)}{r_{rms}} \cdot \left\{ r_{rms} \cdot \omega - \frac{{r_{rms}}^3}{r(t)^2} \cdot \omega + \frac{{r_{rms}}^2}{r(t)^2} \cdot W_{\theta,rms} \right\} \quad (146)$$

- The absolute tangential velocities are revealed by Equation 144

4.8.2 Fan Face to Blade FOD Trajectory/Impact Model

With a model for kinetics that includes in-engine flow dynamics and thermodynamics, the context of turbomachinery in the path of the moving particle has to be considered. Are there conditions favorable to ingestion across the fan? Does the particle trajectory waver significantly due to tangential flow influence? If the particle clears the turbomachinery forward of the 1st HPC rotor, the component of key interest for this thesis, at what velocities, on what side of the airfoil does the particle impact? If indeed there is an impact what penetration depth does it cause, what is the rotor stress at the point of impact and how does this induced flaw affect growth of cracks emanating from that impact site? Figure 4-37 evinces a very tortuous path to impact a blade on the 1st HPC rotor – the particle can impact the fan, the stator or the IGV before it travels to the 1st HPC rotor – the figure also evinces that there are several locations per component where impact can occur.

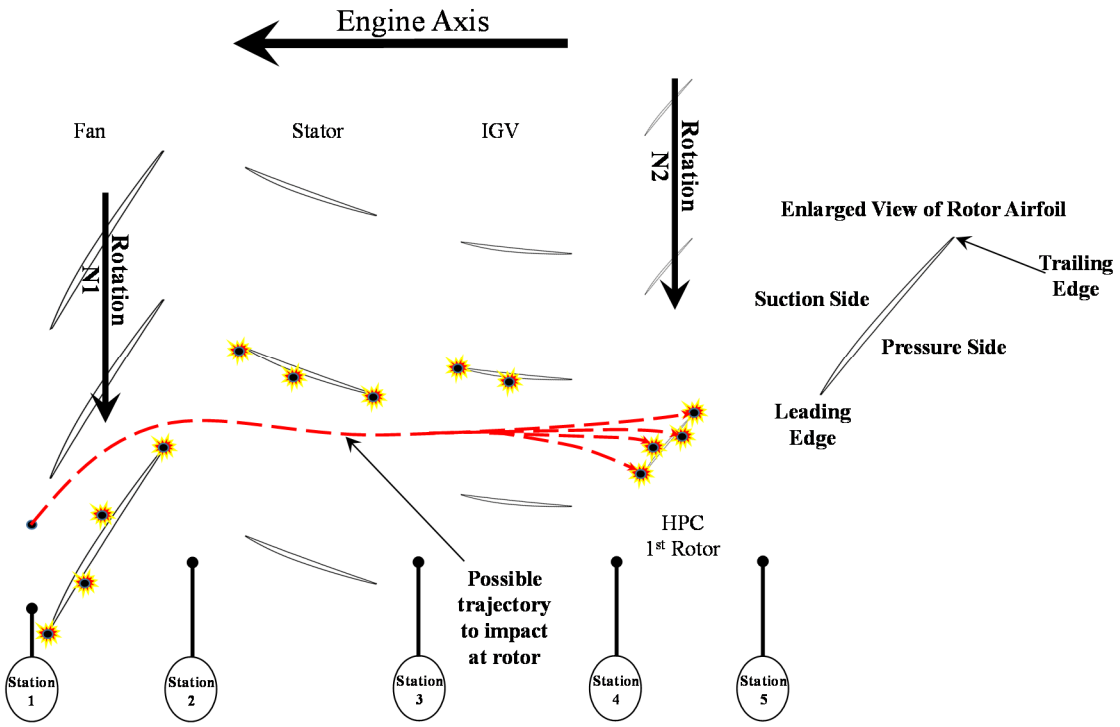


Figure 4.37: Possible Particle Trajectories and Impact Sites

4.8.2.1 Model Assumptions

The fan face to 1st HPC rotor trajectory to impact model works off of a set of assumptions:

- Ricochet is negligible – no modelling is done post ricochet
    - If a particle impacts rotating turbomachinery forward of the bypass section on the Suction Side of the airfoil it is slowed down and spun out to the bypass and out the engine – if the particle is struck by the Pressure Side of the airfoil (this is probable as the angular velocity of the spools where the rotor hubs are attached is very high) the local wheel speed (tangential) may be significantly higher than the axial velocity of the particle, striking the particle like a paddle to the bypass and out of the engine

- It is also assumed that if a particle impacts stationary turbomachinery forward of the bypass section the airfoil it is slowed down and spun out to the bypass and out the engine

- Impact event is very short – only a small portion of momentum is transferred at the impact site – this assumption is consistent with the definition of Hard Body FOD supported in the literature and presented in Section 1.2.1.2 [17]
- The resultant, normal (locally perpendicular to local tangent at point of impact) component of velocity at impact is key and drives modeling of impact depth – the normal impact configuration causes the deepest impact depth [88]
- A statistical relationship exists between impact velocity and depth of penetration – Hambrick's work presents the results of impact tests with varying particle diameters, impact velocities and ensuing impact crater geometry via means and standard deviations – that format is leveraged for data from other researchers who did impact tests and published results [90, 91, 92, 118]

4.8.2.2 Lay-out and Sizing of Turbomachinery

The lay-out and the sizing of the turbomachinery are presented visually in Figure 4-38. The distance between the components is calculated from an approximation of 20% the width of the largest chord projection to the engine centerline axis. Note that span of the rotors at the fan and 1st HPC compressor stage and the span of the stators and IGV are sized by the Meanline Analysis tool (detailed in Section 4.7). Also recall the chord and Stagger Angle at r_{rms} are calculated by the Meanline Analysis tool – the twist and chord at any point radially must be calculated to more accurately model a "real world" rotor with a given airfoil, twist and chord.

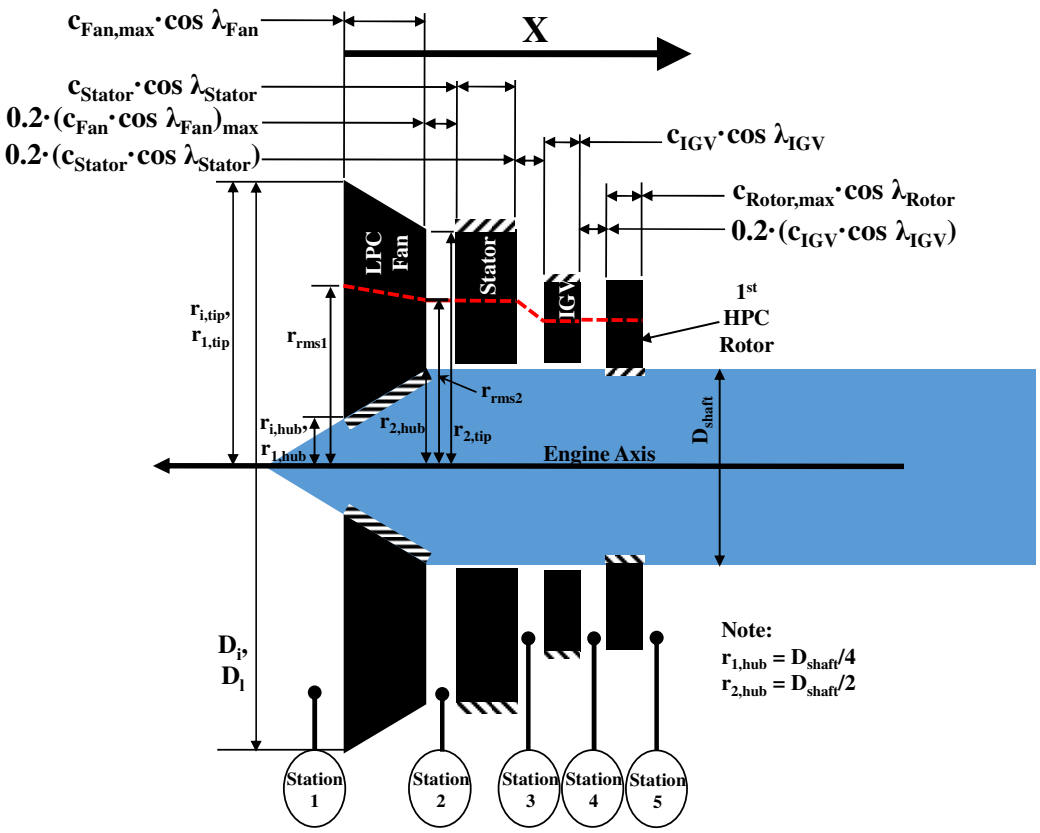


Figure 4.38: Turbo Machinery Lay-out and Sizing

4.8.2.3 Airfoil Fixed Geometry

The airfoil that was chosen for all the components modeled (fan, stator, IGV and 1st HPC rotor blades) was the NACA 65210 - it was chosen because it was employed extensively in legacy axial compressors and ample data is available for it [110, 111] – refer to Figure 4-30 as a guide to match key terminology. The website http://airfoiltools.com provides ample information and the feature of plotting the coordinates of the airfoil surface, chamber line and chord line [119]. The airfoil surface is given in the format of 51 coordinate locations in an XY plane, Figure 4-39 presents the airfoil surface, camber line and chord line – thickness, camber and other airfoil properties are inherent in the coordinates, e.g. the tangent from the leading and trailing edges was employed to calculate the stagger angle of the blade (see Section 4.7) – the figure is not drawn to scale to allow

features such as camber and shape ready discernment. The In-Engine Particle Kinetics simulation code developed with Wolfram Mathematica® 10 presented in Appendix E contains the fan, stator, IGV and 1st HPC rotor coordinates at r_{rms} (accounting for Stagger Angle) of the 51 nodes of the NACA 65-210 airfoil.

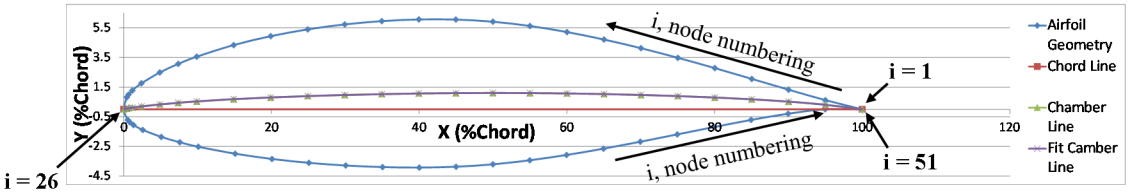


Figure 4.39: NACA 65-210 Airfoil Geometry

4.8.2.4 Airfoil Taper and Twist

While the stator and IGV are fixed in stagger angle and chord not so for the fan and 1st HPC rotor blades – the author of this thesis did not find official, detailed taper and twist data for the Pratt and Whitney F100 engine compressor section - from public domain images the assumption was made that for low bypass, military engines the compressor turbomachinery do not taper or twist as radically as commercial engines do (see Figure 4-40)(it is from this image that a constant shaft diameter of 0.42 m was estimated).

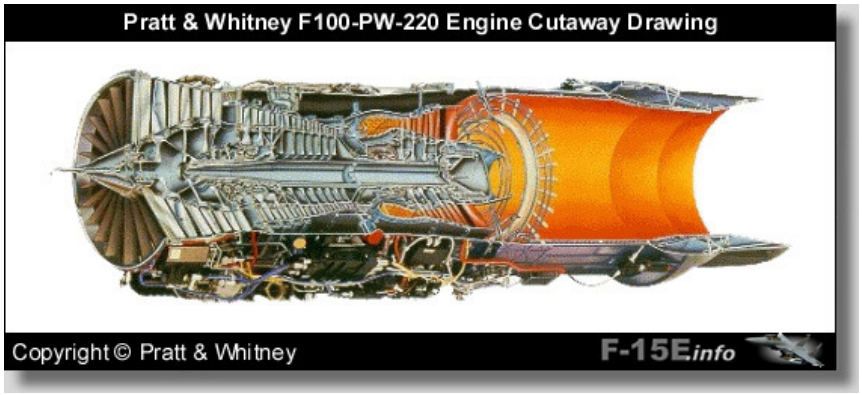


Figure 4.40: F100-PW-220 Cutaway

The NACA 65-210 airfoil raw coordinates are plotted on an XY plane whose X axis is lies on top and parallel to the chord line – Figure presents the transformations that

are be made to provide airfoils along the radial axis scale and twist with respect to the fixed values at r_{rms}:

- XY plane to X'Y' plane that has X' parallel to stagger angle $\lambda_{stagger}$

$$X' = X \cdot \cos\lambda_{stagger} - Y \cdot \sin\lambda_{stagger} \quad (147)$$

$$Y' = X \cdot \sin\lambda_{stagger} + Y \cdot \cos\lambda_{stagger} \quad (148)$$

- Twist with respect to stagger angle at r_{rms} can be approximated with a simple linear relationship that assumes the ratio of twist at tip to root is estimated

$$\lambda(r) = \frac{\left(\frac{\lambda_{tip}}{\lambda_{root}}\right)}{r_{tip} - r_{root}} \cdot (r - r_{rms}) \quad (149)$$

All along r twist must be applied,

$$X'_{twisted}(r) = X' \cdot \cos\lambda(\mathrm{r}) - Y' \cdot \sin\lambda(r) \quad (150)$$

$$Y'_{twisted}(r) = X' \cdot \sin\lambda(\mathrm{r}) + Y' \cdot \cos\lambda(r) \quad (151)$$

- Scaling of chord is more involved, but can be done readily by assuming it has a linear relationship about the chord at r_{rms} such as:

$$c(r) = A \cdot r + b \quad (152)$$

The known data points are c_{rms}, r_{rms}, r_{tip} and r_{root},

$$c(r_{rms}) = c_{rms} = A \cdot r_{rms} + b \quad (153)$$

$$c(r_{root}) = c_{root} = A \cdot r_{root} + b \quad (154)$$

$$c(r_{tip}) = c_{tip} = A \cdot r_{tip} + b \quad (155)$$

Assuming that the ratio of chord at tip to root is estimated a-priori, Equation 148 can be transformed to Equation 155a,

$$c(r_{tip}) = c_{root} \cdot \left({}^{c_{tip}}/_{c_{root}}\right) = A \cdot r_{tip} + b \quad (155a)$$

solving for the A slope, intercept b and c_{root} from the three variable, linear system, rearranging terms (recalling what r_{tip} and r_{shaft} are in terms of engine parameters for the fan are $D_l/2$ and $D_{shaft}/2$ respectively and $D_{rotor}/2$ and $D_{shaft}/2$ respectively for the 1st HPC rotor),

$$c(r) = c_{rms} \cdot \left\{ \frac{\left({}^{c_{tip}}/_{c_{root}}\right)\cdot(r - r_{root}) + (r_{tip} - r)}{\left({}^{c_{tip}}/_{c_{root}}\right)\cdot(r_{rms} - r_{root}) + (r_{tip} - r_{rms})} \right\} \tag{152a}$$

Aside: for simulation code for in-engine particle travel Equation 152a is rearranged in terms of a span to chord ratio (H/c) to make the formulation more intuitive,

$$\left({}^{H}/_{c}\right)_{rms} = \frac{\left\{r_{tip} + r_{rms}\cdot\left[\left({}^{c_{tip}}/_{c_{root}}\right) - 1\right] - \left({}^{c_{tip}}/_{c_{root}}\right)\cdot c_{root}\right\}}{c_{rms}\cdot\left[\left({}^{c_{tip}}/_{c_{root}}\right) - 1\right]} \tag{156}$$

$$c(r) = \frac{r - r_{rms}}{\left({}^{H}/_{c}\right)_{rms}} + c_{rms} \tag{152b}$$

- The aggregate formulation for transformation, twist about c_{rms} takes the form:

$$X'_{scaled,twisted}(r) = [X' \cdot \cos\lambda(\mathrm{r}) - Y' \cdot \sin\lambda(r)] \cdot \frac{c(r)}{c_{rms}} \tag{157}$$

$$Y'_{scaled,twisted}(r) = [X' \cdot \sin\lambda(\mathrm{r}) + Y' \cdot \cos\lambda(r)] \cdot \frac{c(r)}{c_{rms}} \tag{158}$$

4.8.2.5 Airfoil Section Stress

The airfoil is subjected to several modes of loading due to the angular velocity of the rotor and the localized mass a distance r from the engine centerline axis – axial, bending, torque. These quasi-static modes of loading give rise to section stresses at each airfoil section a distance r away from the engine centerline axis. Only axial stresses will be considered for this thesis for the following reasons:

- Regardless of the location of the point about which an airfoil is twisted, the resultant tensile force can be transferred to a centroid by means of a force-

couple system (an equivalent force and moment) – that again reveals a tensile force that produces a tensile stresses (see left hand side of Figure 4-41) – the assumption of largely uniaxial stress is corroborated from Finite Element Analysis (FEA) stress profiles (see right hand side of Figure 4-41)

- In a later section a model for growing cracks out of the impact locations will be presented – that model relies on geometric properties that affect the growth of cracks, these properties (called β geometry factors) are much more readily available in the public domain for tensile stresses than for other types of stresses – the bending mode of loading is specifically complex because twisted shapes couple in bending generating a complex mix of tensile and compressive stresses, thus axial stresses will be pursued
- As will be shown in this section, a rare analytical, closed form solution can be had for the axial stresses at an r away from the datum – the author of this thesis derived the analytical formulation – these stresses are to be termed reference stresses, the term is consequential for growing cracks

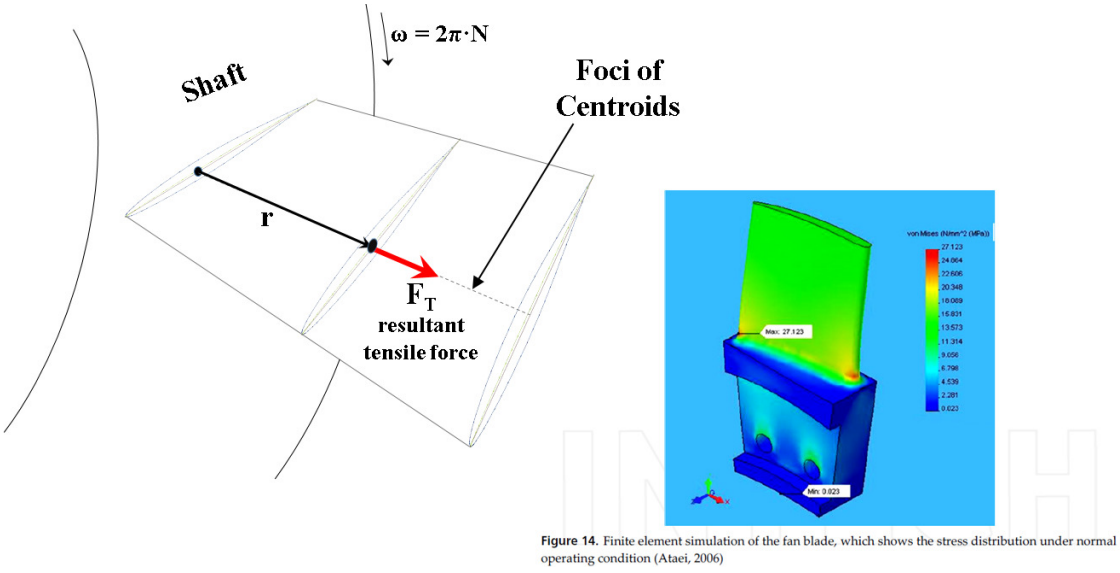


Figure 4.41: Tensile Forces at Airfoil Section

From continuum mechanics tensile forces at a point can be derived from the rate of differential extensile force and loading (rotational in this case) equilibrium [120]:

- Differential equilibrium,

$$\frac{d}{dr}\left[E \cdot A(r) \cdot \frac{du}{dr}\right] = -\rho_{material} \cdot A(r) \cdot \omega^2 * r \tag{159}$$

Where,

E – material Young's modulus

A(r) – airfoil section area

$\rho_{material}$ – airfoil material

ω – angular velocity

u – displacement along r

- Recall that Equations 157 and 158 provide a means to transform, twist and scale any airfoil coordinate with respect to r_{rms} – but as was stated before, only tensile loading is considered, thus twisting the airfoil has no bearing on the calculation of stresses, thus the equations are modified by setting λ(r) to zero as follows:

$$X'_{scaled}(r) = [X'] \cdot \frac{c(r)}{c_{rms}} \tag{157a}$$

$$Y'_{scaled}(r) = [Y'] \cdot \frac{c(r)}{c_{rms}} \tag{158a}$$

- It is proposed that the area of the airfoil section can be approximated by trapezoidal integration, keeping in mind that there are 51 coordinate points numbered per Figure 4-39:

$$A(r) = \sum_{i}^{25}\left\{\left[\frac{Y'_{scaled}[r,i+1]+Y'_{scaled}[r,i]}{2}\right] \cdot \left[X'_{scaled}[r,i] - X'_{scaled}[r,i+1]\right] - \left[\frac{Y'_{scaled}[r,52-i]+Y'_{scaled}[r,51-i]}{2}\right] \cdot \left[X'_{scaled}[r,52-i] - X'_{scaled}[r,51-i]\right]\right\} \tag{160}$$

Aside: Time determines the location of the particle, including in the radial direction. Because the stresses being considered are quasi static, time becomes of a scale larger than that of particle kinetics – thus r(t) becomes just r.

- Recall that in the formulation for $X'_{scaled}(r)$ and $Y'_{scaled}(r)$ there is a scaling factor $c(r)/c_{rms}$ (Equation 152b) – that term can be extracted from Equation 160 as it is the only one dependent on r, thus revealing A(r) in a convenient form:

$$A(r) = \left(\frac{c(r)}{c_{rms}}\right)^2 \cdot \sum_i^{25} \left\{\left[\frac{Y\prime[r,i+1]+Y\prime[r,i]}{2}\right] \cdot \left[X'[r,i] - X'[r,i+1]\right] - \left[\frac{Y\prime[r,52-i]+Y\prime[r,51-i]}{2}\right] \cdot \left[X'[r,52-i] - X'[r,51-i]\right]\right\} \quad (160a)$$

- Taking the derivative on the left hand side of Equation 159 with r and assuming the material is isotropic:

$$\frac{d}{dr}\left[E \cdot A(r) \cdot \frac{du}{dr}\right] = E \cdot \left\{\frac{dA(r)}{dr} \cdot \frac{du}{dr} + A(r) \cdot \frac{d^2u}{dr^2}\right\} = -\rho_{material} \cdot A(r) \cdot \omega^2 \cdot r \quad (159a)$$

- The X'Y' terms in Equation 160 are not functions of r, thus they drop out of Equation 159a, revealing a very simple form:

$$E \cdot \left\{\frac{2}{(H/c)_{rms} \cdot c_{rms}} \cdot \left(\frac{c(r)}{c_{rms}}\right) \cdot \frac{du}{dr} + \left(\frac{c(r)}{c_{rms}}\right)^2 \cdot \frac{d^2u}{dr^2}\right\} = -\rho_{material} \cdot \left(\frac{c(r)}{c_{rms}}\right)^2 \cdot \omega^2 \cdot r \quad (159b)$$

- Equation 159b is solved analytically through Wolfram Mathematica® 10 with the following boundary conditions:

  No displacement at root,

$$u(r = r_{root}) = 0 \quad (161)$$

  No axial force at the tip (blade ends),

$$E \cdot A\left(r = r_{tip}\right) \cdot \frac{du(r=r_{tip})}{dr} = 0 \tag{162}$$

Resulting in axial displacement,

$$u(r) = \ ((2 \cdot \mathrm{r}_{root} - 2 \cdot \mathrm{r}) \cdot (-3 \cdot (2 \cdot \mathrm{r}_{tip})^4 + 16 \cdot (2 \cdot \mathrm{r}_{tip})^3 \ (-\left({}^{H}/_{c}\right)_{rms} \cdot c_{rms} + \mathrm{r}_{rms}) - 24 \cdot (2 \cdot \mathrm{r}_{tip})^4 \ (-\left({}^{H}/_{c}\right)_{rms} \cdot c_{rms} + \mathrm{r}_{rms})^2 + 2 \cdot (4 \cdot \left(\left({}^{H}/_{c}\right)_{rms} \cdot c_{rms}\right)^2 \ ((2 \cdot \mathrm{r}_{root})^2 + 2 \cdot (2 \cdot \mathrm{r}_{root}) \cdot \mathrm{r} + 4 \cdot \mathrm{r}^2) + (2 \cdot \mathrm{r}_{root}) \cdot \mathrm{r} \cdot ((2 \cdot \mathrm{r}_{root})^2 + 2 \cdot (2 \cdot \mathrm{r}_{root}) \cdot \mathrm{r} + 4 \cdot \mathrm{r}^2) - ((2 \cdot \mathrm{r}_{root}) + 2 \cdot \mathrm{r})^3 \cdot \mathrm{r}_{rms} + 4 \cdot ((2 \cdot \mathrm{r}_{root})^2 + 2 \ (2 \cdot \mathrm{r}_{root}) \cdot \mathrm{r} + 4 \cdot \mathrm{r}^2) \ \cdot {\mathrm{r}_{rms}}^2 + \left({}^{H}/_{c}\right)_{rms} \cdot c_{rms} \cdot (((2 \cdot \mathrm{r}_{root}) + 2 \cdot \mathrm{r})^3 - 8 \cdot \ ((2 \cdot \mathrm{r}_{root})^2 + 2 \cdot \ (2 \cdot \mathrm{r}_{root}) \cdot \mathrm{r} + 4 \ \cdot \mathrm{r}^2) \cdot \mathrm{r}_{rms}))) \cdot \omega^2 \cdot \rho_{material})/(192 \cdot \mathrm{E} \cdot ((2 \cdot \mathrm{r}_{root}) + 2 \cdot \left({}^{H}/_{c}\right)_{rms} \cdot c_{rms} - 2 \cdot \mathrm{r}_{rms}) \ (\left({}^{H}/_{c}\right)_{rms} \cdot c_{rms} + \mathrm{r} - \cdot \mathrm{r}_{rms})) \tag{163}$$

Recall from elementary mechanics that the axial force N_1 (not N1 the fan spool speed) is as follows,

$$N_1 = E \cdot A(r) \cdot \frac{du}{dr} \tag{164}$$

The formulation for axial stress is as follows,

$$\sigma_1 = \frac{N_1}{A(r)} = \ (1/(192 \cdot \mathrm{E} \cdot (\left({}^{H}/_{c}\right)_{rms} \cdot c_{rms} + \mathrm{r} - \mathrm{r}_{rms})^2))(3 \cdot (2 \cdot \mathrm{r}_{tip})^4 + 16 \cdot (2 \cdot \mathrm{r}_{tip})^3 \ (\left({}^{H}/_{c}\right)_{rms} \cdot c_{rms} - \mathrm{r}_{rms}) + 24 \cdot (2 \cdot \mathrm{r}_{tip})^2 \cdot (-\left({}^{H}/_{c}\right)_{rms} \cdot c_{rms} + \mathrm{r}_{rms})2 - 16 \cdot \mathrm{r}^2 \cdot (6 \cdot (\left({}^{H}/_{c}\right)_{rms} \cdot c_{rms})^2 + 3 \cdot \mathrm{r}^2 + 4 \cdot \left({}^{H}/_{c}\right)_{rms} \cdot c_{rms} \cdot (2 \cdot \mathrm{r} - 3 \cdot \mathrm{r}_{rms}) - 8 \cdot \mathrm{r} \cdot \mathrm{r}_{rms} + 6{\mathrm{r}_{rms}}^2)) \cdot \omega^2 \cdot \rho_{material} \tag{165}$$

Aside: the σ_1 stress is the single principal stress, if failure was not by HCF, it would be the sole failure criteria, if no additional failure modes were considered.

4.8.2.6 Impact Law

If the distance between a particle and the closest node on a blade section is less than or equal to the radius of a FOD particle impact has occurred. This law is presented visually by Figure 4-42 and mathematically by Equation 166 – the equation is a reminder that the impact problem is fully 3-D.

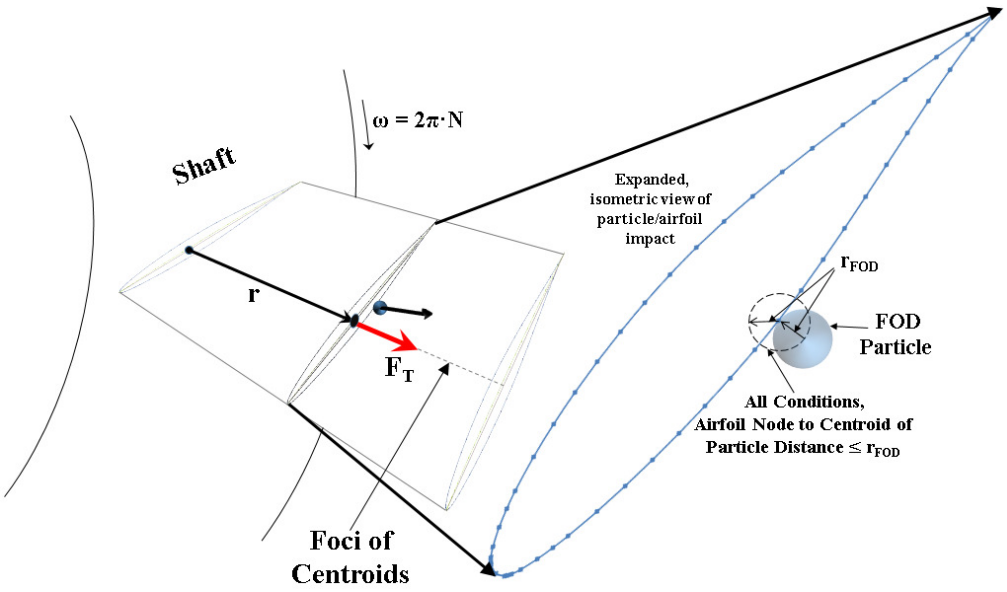


Figure 4.42: Visual Representation of Impact Law

$$D_{FODParticle/AirfoilNode} =$$

$$\left(\left(X_{particle}(t) - X_{airfoilnode}(t,i,j)\right)^2 + \left(Y_{particle}(t) - Y_{airfoilnode}(t,i,j)\right)^2 + \left(Z_{particle}(t) - Z_{airfoilnode}(t,i,j)\right)^2 \right)^{\frac{1}{2}} \leq r_{FOD} \qquad (166)$$

Where,

t – denotes time in seconds, begun at the arrival of the FOD particle at the fan face

i – airfoil nodes (51 coordinate nodes on NACA 65-210 airfoil section)

j - accounts for blade that had closest node to FOD particle – the fan hub has 34 blades while the 1st HPC rotor hub has 38 blades

The particle and blade coordinates of Equation 166 must be calculated from the particle kinetics formulations detailed in Section 4.8.1, Section 4.8.2.1 and Section 4.8.2.4. Particle coordinates can be readily calculated from Figure 4-35 by applying the following relationships of axial and planar displacement:

- Axial displacement of the particle located from its datum at fan face/inlet is the solution to X(t) from Equation 142:

  $$X_{particle}(t) = \mathrm{X(t)} \tag{167}$$

- With an YZ plane located at the engine centerline axis at the inlet (see Figure 4-43), $Y_{particle}$ and $Z_{particle}$ are as follows:

  $$Y_{particle}(t) = -r(t) \cdot \sin\theta(t) \tag{168}$$

  $$Z_{particle}(t) = -r(t) \cdot \cos\theta(t) \tag{169}$$

The coordinates of the airfoil blade nodes can be calculated from Figure 4-43 – specifically, in the figure dashed lines along a radial direction connect engine nodes from airfoil sections, revealing that sections displace in rigid body angular motion as follows:

- The axial displacement, also referenced from the datum of the engine centerline axis at inlet, of the nodes on an airfoil section is only dependent on time and on node location (see Figure 4-39) - the formulation will be the time and node location (i counter) dependent expanded version of Equation 157:

  $$X_{airfoilnode}(t,i) = X'_{scaled,twisted}(t,i) = \Big[\big(X[i] \cdot \cos\lambda_{stagger} - Y[i] \cdot \sin\lambda_{stagger}\big) \cdot \cos\lambda\big(\mathrm{r(t)}\big) - \big(X[i] \cdot \sin\lambda_{stagger} + Y[i] \cdot \cos\lambda_{stagger}\big) \cdot \sin\lambda\big(r(t)\big)\Big] \cdot \frac{c(r(t))}{c_{rms}} \tag{157b}$$

- The planar displacement, referenced from angular location when the particle arrives at the inlet or when it arrives at the 1st HPC rotor, of nodes on an airfoil are dependent on time, node location and which blade is being considered (j counter, recall that a spool hub will have several blades or rotors) – corrections are embedded to account for the rotation of blades whose angular location is evenly/randomly distributed when the simulation time begins and are angularly spaced apart (angular pitch) – the formulation will translate Equation 158 for $Y'_{scaled,twisted}(r)$ in the 2-D plane of an airfoil into the time, node location and blade reference engine centerline axis components Y and Z (see Figure 4-43):

$$Y_{airfoilnode}(t,i,j) = \left\{\left[\left(X[i]\cdot\cos\lambda_{stagger} - Y[i]\cdot\sin\lambda_{stagger}\right)\cdot\sin\lambda(\mathrm{r}) + \left(X[i]\cdot\sin\lambda_{stagger} + Y[i]\cdot\cos\lambda_{stagger}\right)\cdot\cos\lambda(r)\right]\cdot\frac{c(r)}{c_{rms}}\right\}\cdot\cos\left[\omega\cdot t + (0.5 - EvenlyDistributedRandNumb)\cdot\frac{2\cdot\pi}{n_{blades}} + (j-1)\cdot\frac{2\cdot\pi}{n_{blades}}\right] - r(t)\cdot\sin\left[\omega\cdot t + (0.5 - EvenlyDistributedRandNumb)\cdot\frac{2\cdot\pi}{n_{blades}} + (j-1)\cdot\frac{2\cdot\pi}{n_{blades}}\right] \quad (170)$$

$$Z_{airfoilnode}(t,i,j) = \left\{\left[\left(X[i]\cdot\cos\lambda_{stagger} - Y[i]\cdot\sin\lambda_{stagger}\right)\cdot\sin\lambda(\mathrm{r}) + \left(X[i]\cdot\sin\lambda_{stagger} + Y[i]\cdot\cos\lambda_{stagger}\right)\cdot\cos\lambda(r)\right]\cdot\frac{c(r)}{c_{rms}}\right\}\cdot\sin\left[\omega\cdot t + (0.5 - EvenlyDistributedRandNumb)\cdot\frac{2\cdot\pi}{n_{blades}} + (j-1)\cdot\frac{2\cdot\pi}{n_{blades}}\right] - r(t)\cdot\cos\left[\omega\cdot t + (0.5 - EvenlyDistributedRandNumb)\cdot\frac{2\cdot\pi}{n_{blades}} + (j-1)\cdot\frac{2\cdot\pi}{n_{blades}}\right] \quad (171)$$

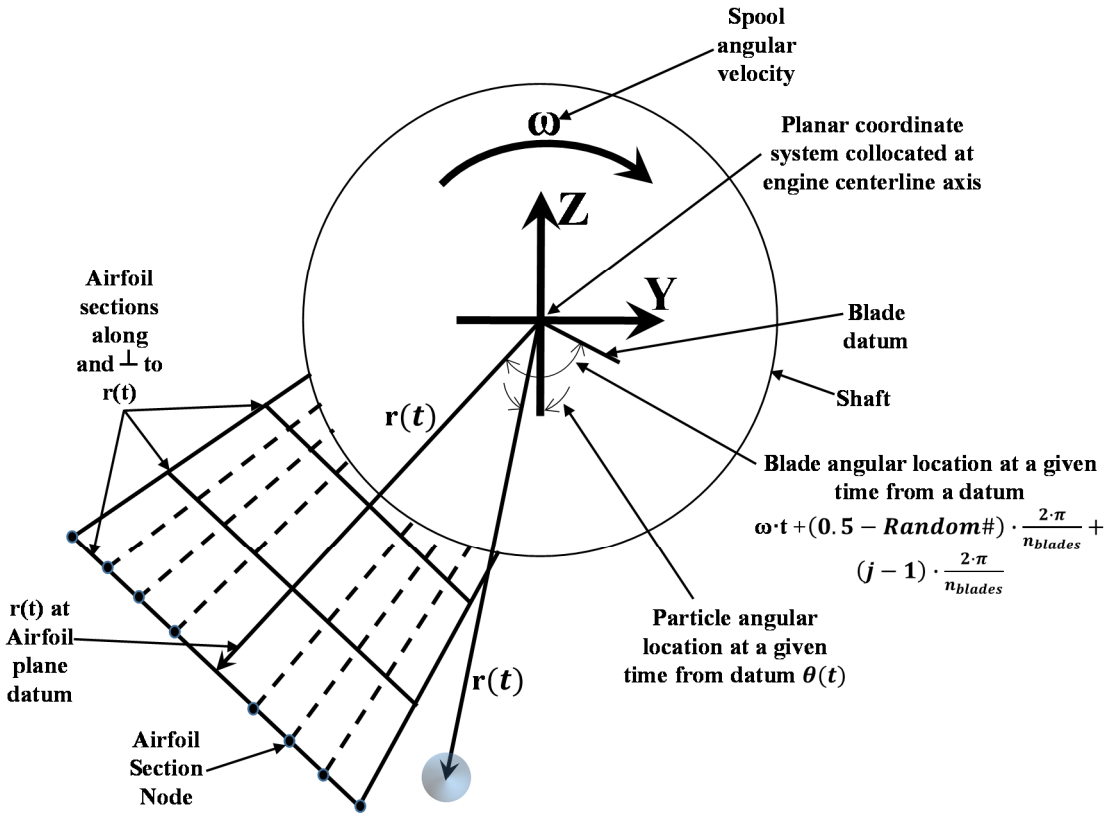


Figure 4.43: Particle & Airfoil Nodes Locations

4.8.2.7 Impact Velocities

After the condition for impact (Equation 166) is met the velocities at which the particle impacted a blade section can be calculated from the dynamics of the airfoil and the kinetics of the particle. Figure 4-44 depicts the FOD particle and airfoil node velocities on the plane of the airfoil section. V_x is the velocity of the particle in the local airfoil coordinate system that also coincides with the global X axis referenced from the engine centerline axis – V_x is the solution for $\frac{dX}{dt}$ from Equation 142. The local airfoil plane particle velocity V_y is comprised of the global YZ plane (referenced form the engine centerline axis at the inlet) components. The impact or incident velocities differ on the Suction vs. Pressure Side of the airfoil (see Figure 4-44).The formulation is as follows:

- For the Suction Side of the airfoil:

Taking the time derivative (with chain rule consideration where appropriate) of the coordinate formulations from Equations 167, 168 and 169:

$$V_x = \dot{X}_{particle}(t) = \dot{\mathrm{X}}(\mathrm{t}) \tag{172}$$

$$\dot{Y}_{particle}(t) = \frac{d(-r(t)\cdot\sin\theta(t))}{dt} = -\left(r(t)\cdot\cos\theta(t)\cdot\frac{d\theta(t)}{dt} + \sin\theta(t)\cdot\frac{dr(t)}{dt}\right) \tag{173}$$

$$\dot{Z}_{particle}(t) = \frac{d(-r(t)\cdot\cos\theta(t))}{dt} = \left(r(t)\cdot\sin\theta(t)\cdot\frac{d\theta(t)}{dt} - \cos\theta(t)\cdot\frac{dr(t)}{dt}\right) \tag{174}$$

Recognizing that the airfoil section, planar particle velocity V_y is the same on any blade on the spool hub,

$$V_y = \dot{Z}_{particle}(t)\cdot\cos\theta(t) + \dot{Y}_{particle}(t)\cdot\sin\theta(t) \tag{175}$$

The local node tangent angle ϕ is required to calculate the velocity components normal at the local normal vector at the point of impact – it can be estimated from the airfoil coordinates,

$$\phi_{SuctionSide} = \tan^{-1}\left\{\frac{Y[i]-Y[i+1]}{X[i]-X[i+1]}\right\} \tag{176}$$

After accounting for the local wheel speed component, and recalling that all velocities have been projected onto the plane of the airfoil (any $V_z \approx 0$ m/s) the impact velocity on the Suction Side of the airfoil becomes,

$$V_{impact,SuctionSide} = \sqrt{(V_x\cdot\cos\phi_{SuctionSide})^2 + {V_z}^2} \tag{178}$$

- For the Pressure Side of the airfoil:

  Once again the local node tangent angle ϕ is required to calculate the velocity components normal at the local normal vector at the point of impact – it requires a modification as the node count on the Pressure Side travels on the opposite side as that of the Suction Side (see 4-39),

$$\phi_{PressureSide} = \tan^{-1}\left\{\frac{Y[i+1]-Y[i]}{X[i+1]-X[i]}\right\} \tag{179}$$

Once again, accounting for the local wheel speed component, and recalling that all velocities have been projected onto the plane of the airfoil (any $V_z \approx 0$ m/s) the impact velocity on the Pressure Side of the airfoil becomes,

$$V_{impact,PressureSide} = \sqrt{\left\{\left(V_y + r(t)\cdot\omega\right)\cdot \sin\phi_{PressureSide}\right\}^2 + V_z{}^2} \qquad (180)$$

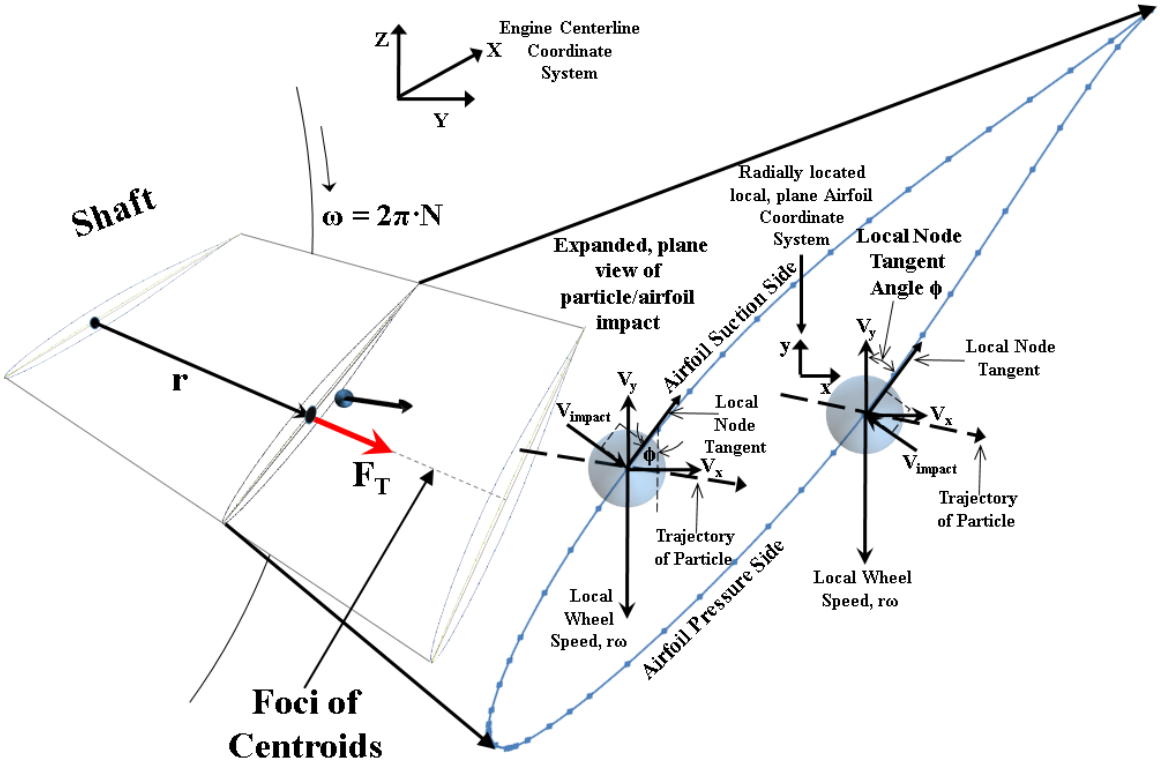


Figure 4.44: FOD Particle/Blade Node Velocities

The In-Engine Particle Kinetics simulation code developed with Wolfram Mathematica® 10 presented in Appendix E contains the entire formulation and structure that was presented in this section – the intricacies of Mathematica® 10 are better appreciated by thoroughly reviewing the code.

4.8.2.8 Impact Velocity to Particle Penetration

As mentioned previously a statistical relationship exists between impact velocity and depth of penetration – Hambrick's work presents the results of impact tests with

varying particle diameters, impact velocities and ensuing impact crater geometry via means and standard deviations – that format is leveraged for data from other researchers who performed impact tests and published results [90, 91, 92, 118]. Data that relates impact velocities to the most common sizes of Hard FOD particles (a recent study at four military bases and two research stations in the UK revealed that 87.6 % of the debris collected on the runways was Hard Body FOD smaller than 5 mm in diameter [93], see Table 4.4) is sparse – two particle sizes of mild carbon steel (density 7850 kg/m^3) of 1.33 and 3.2 mm diameter contained sufficient spread in their relationships of impact velocity to penetration depth to be usable for this thesis [90, 92, 118] – in a later section a case study will reveal that impact velocities do vary significantly. The blade material for these two particle sizes was the two phase (α+β), solution treated, over aged (SOA) Ti–6Al–4V alloy (with chemical composition Ti–6.30Al–4.19V–0.19Fe–0.19O, %wt.) [83] - the material is extensively employed for low temperature compressor applications. The raw data, along with statistical measures for these two particle sizes is presented in Table 4.6. The process by which impact depth is extracted from the statistical measures is provided in Figure 4-45. As the impactor is considered infinitely stiff as compared to material being impacted the impact depth has a relationship with impact diameter depicted in Figure 4-46 and Equation 181 – this assumption, often called out in the literature as twice the length of crack on the surface or 2·c, is supported by impact test photographic evidence and computational modeling on same material coupons, neglecting the small rim material pile-up (see Figures 4-47 and 4-48) [84, 92]. The In-Engine Particle Kinetics simulation code presented in Appendix E contains the Mathematica® 10 functions to achieve the process and extraction of impact penetration depths and widths.

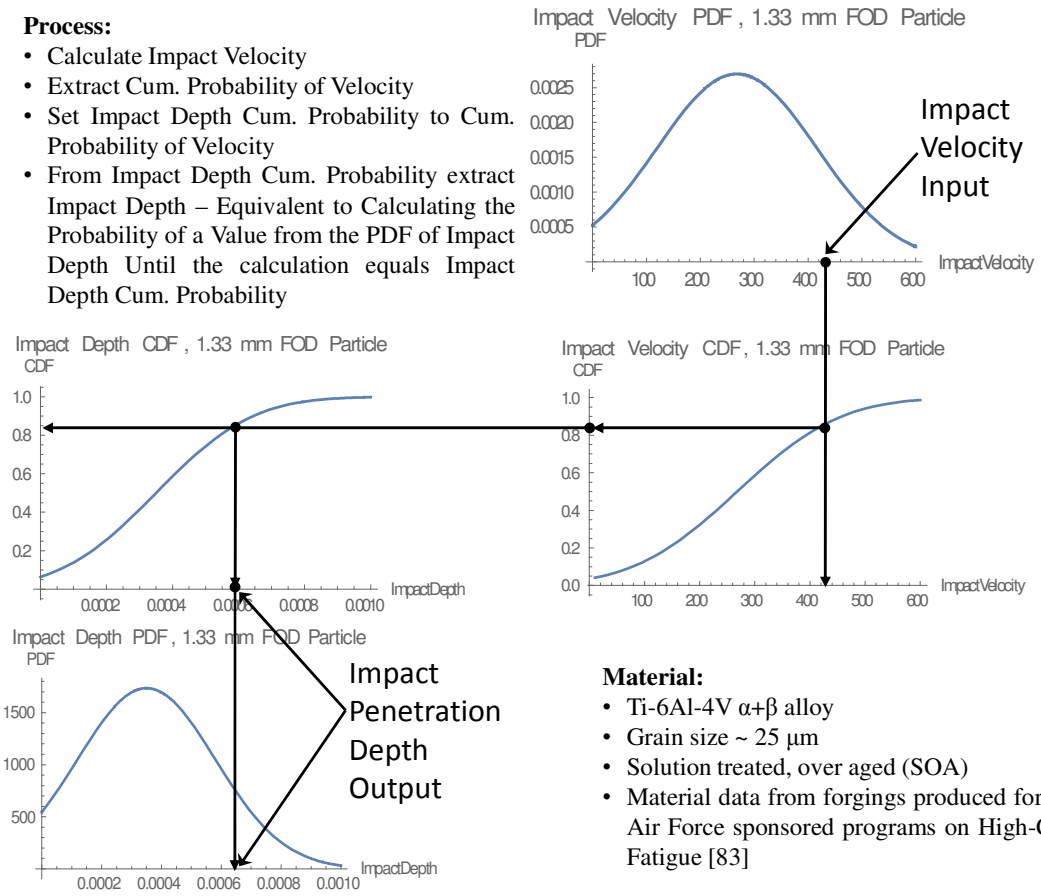
Table 4.6: Impact Velocity to Penetration Depth Data and Statistics

Indentor Diameter (m)	Impact Velocity Normal (m/s)	Impact Depth (m)	Impact Radius (m)
0.00133	264.13775	0.00026	0.00053
0.00133	264.13775	0.00032	0.00057
0.00133	264.13775	0.00034	0.00058
0.00133	264.13775	0.00035	0.00059
0.00133	264.13775	0.00042	0.00062
0.00133	264.13775	0.00042	0.00062
0.00133	264.13775	0.00027	0.00053
0.00133	264.13775	0.00026	0.00053
0.00133	264.13775	0.00038	0.00060
0.00133	58.02370	0.00004	0.00023
0.00133	58.02370	0.00006	0.00028
0.00133	58.02370	0.00005	0.00026
0.00133	58.02370	0.00003	0.00021
0.00133	58.02370	0.00005	0.00024
0.00133	448.60116	0.00054	0.00065
0.00133	448.60116	0.00056	0.00066
0.00133	448.60116	0.00060	0.00066
0.00133	448.60116	0.00062	0.00066
0.00133	448.60116	0.00066	0.00066
0.00133	448.60116	0.00078	0.00067
0.00315	200.00000	0.00041	0.00102
0.00315	200.00000	0.00036	0.00104
0.00320	300.00000	0.00067	0.00134
0.00320	300.00000	0.00067	0.00134
0.00320	200.00000	0.00043	0.00107
0.00320	200.00000	0.00043	0.00107
0.00320	259.80762	0.00067	0.00129
0.00320	259.80762	0.00067	0.00129
0.00320	173.20508	0.00034	0.00097
0.00320	173.20508	0.00034	0.00097

Statistical Measures	
Diameter (m)	0.00133
Mean Normal Velocity (m/s)	**STD Normal Velocity (m/s)**
267.94826	148.01940
Mean Normal Impact Depth (m)	**STD Normal Impact Depth (m)**
0.00035	0.00023
Diameter (m)	0.0032
Mean Normal Velocity (m/s)	**STD Normal Velocity (m/s)**
226.6025404	48.89090684
Mean Normal Impact Depth (m)	**STD Normal Impact Depth (m)**
0.0004982	0.000151357

Process:

- Calculate Impact Velocity
- Extract Cum. Probability of Velocity
- Set Impact Depth Cum. Probability to Cum. Probability of Velocity
- From Impact Depth Cum. Probability extract Impact Depth – Equivalent to Calculating the Probability of a Value from the PDF of Impact Depth Until the calculation equals Impact Depth Cum. Probability



Material:

- Ti-6Al-4V α+β alloy
- Grain size ~ 25 μm
- Solution treated, over aged (SOA)
- Material data from forgings produced for U.S. Air Force sponsored programs on High-Cycle Fatigue [83]

Figure 4.45: Impact Velocity to Penetration Process

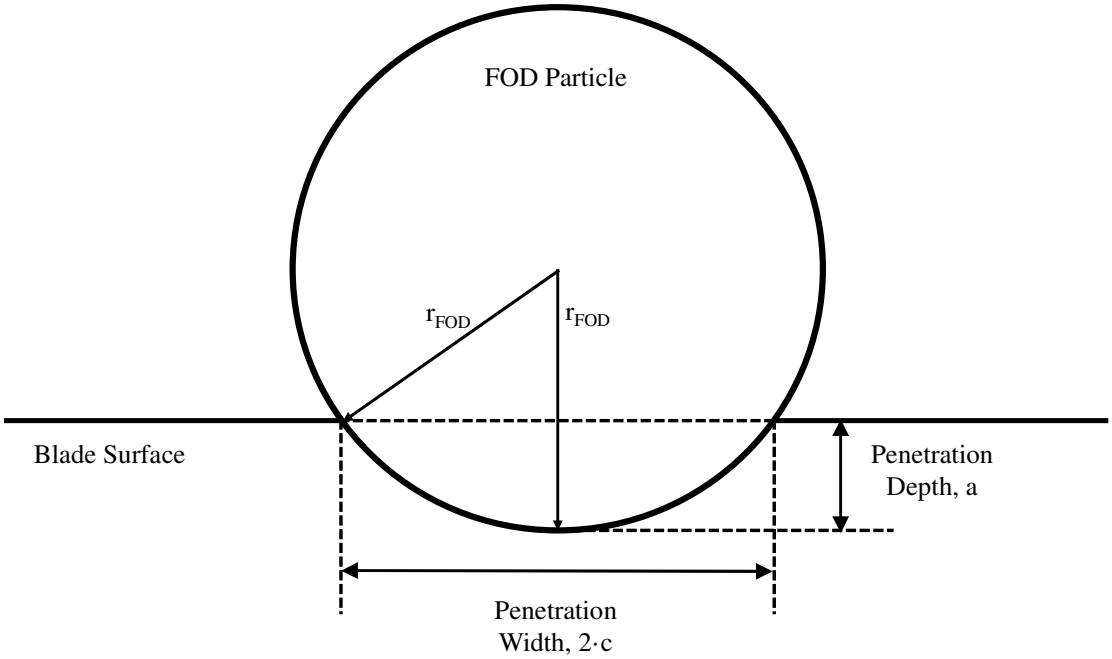


Figure 4.46: Penetration Depth to Width Relationship

$$PenetrationWidth = 2 \cdot c = 2 * \sqrt{(r_{FOD})^2 - (r_{FOD} - PenetrationDepth)^2} \quad (181)$$

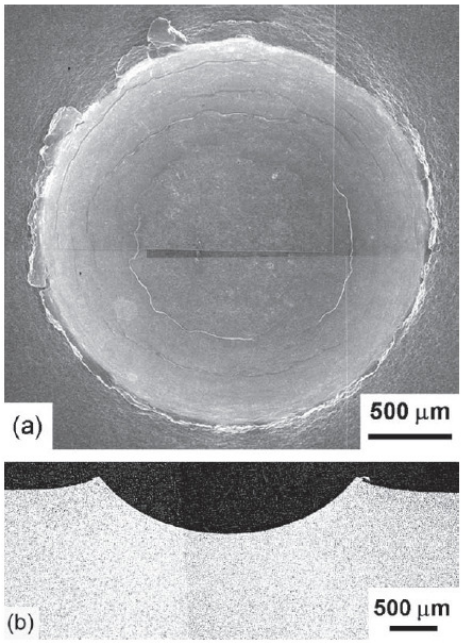


Fig. 10. (a) Scanning electron micrograph and (b) cross-section of impact site after 300 m/s normal (90°) impact using 3.2 mm steel

Figure 4.47: Impact Test Crater & Cross-section of Ti-6Al-4V Specimen

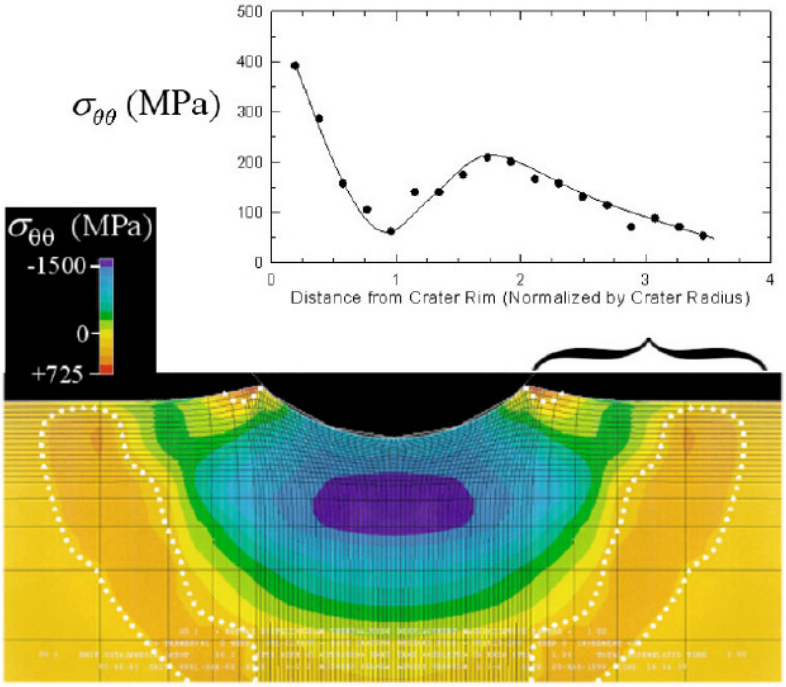


Fig. 3. Finite element prediction (lower contour plot) of the cross-sectional variation and experimental evaluation (upper graph) of the radial variation of residual hoop stresses for a quasi-static indent.

Figure 4.48: FEM, Impact Crater & Cross-section of Ti-6Al-4V Specimen

4.8.2.9 Impact Model Pseudo Code

The key formulae for impact of mild steel particles on Ti-6Al-4V blades has been presented in the previous sections. Additional calculations are also made by the In-Engine Particle Kinetics simulation code (presented in Appendix E): %Chord, impact statistics and visualizations – the reader is encouraged to review the code. The basic workings of the code are presented by the pseudo code in Figure 4-49. Figures 4-50 and 4-51 depict particle impact at the fan and at a 1st HPC rotor blade. Equation 182 presents the basis for the time increment that was employed in the code – this increment is the minimum that ensures that particles do not artificially miss turbomachinery.

```
Do[
    Do[
        Do[
            Distance of FOD Particle to Airfoil Node
            If[Distance of FOD Particle to Airfoil Node ≤ r_FOD,
               Record Particle Impact Location;
               Do[
                    Record All Coordinates of Airfoil Nodes, {ii,1,51}];
               Record Impact Surface (Suction or Pressure Side);
               Calculate/Record %Chord of airfoil plane;
               Calculate/Record r at impact;
               Calculate/Record  Impact Velocity;
               Calculate/Record  Impact Depth/Width;
               Calculate/Record  Blade Thickness at impact site;
               Calculate/Record  Reference Stress at r at impact site;
               Calculate/Record  Impact Velocity;
               Record turbomachinery strikes/misses and bypass escapes,
               {j, 1, n_Blades}],{i, 1, n_AirfoilNodes}],{t, t, t, Δt}]
```

Figure 4.49: In-Engine Particle Kinetics Simulation Pseudocode

$$\Delta t = \frac{r_{FOD}}{\sqrt{{U_{rms,fan}}^2+{V_{inlet,min}}^2}} = \frac{0.0032}{\sqrt{334.96^2+57.2^2}} = 0.00000942\ s\ \approx 0.00001\ s \qquad (182)$$

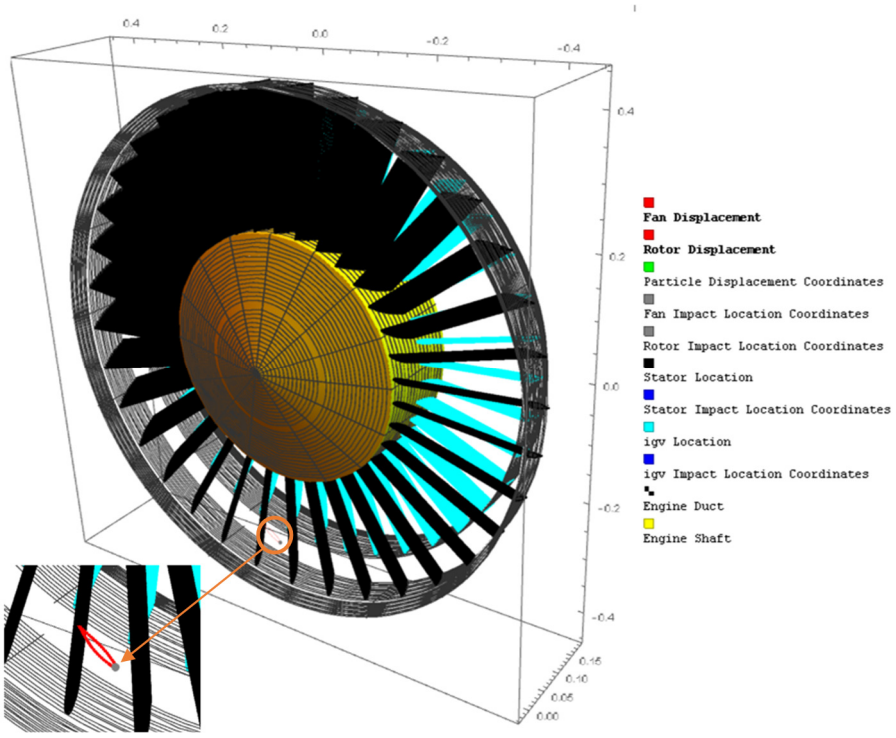


Figure 4.50: 1.33 mm Particle Impact at Fan Blade

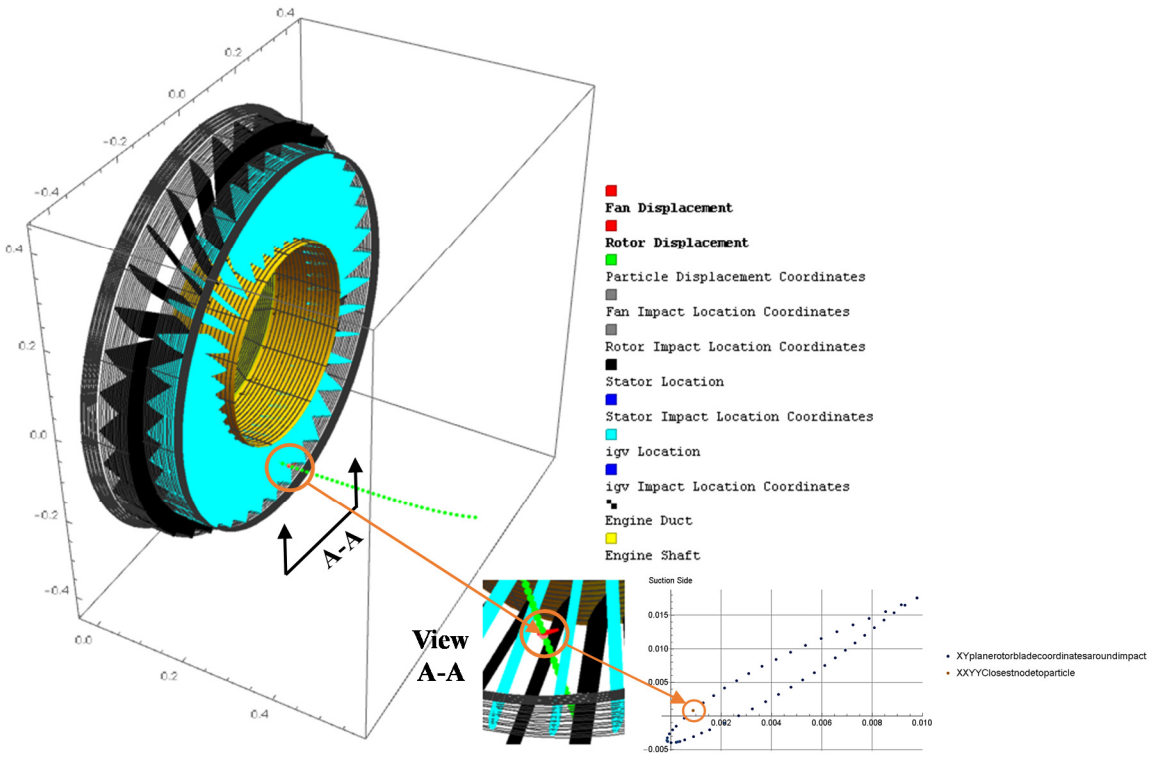


Figure 4.51: 1.33 mm Particle Impact at 1st HPC Rotor Blade

4.9 CRACK GROWTH MODEL

As was previously mentioned, the material that will be considered is the two phase (α+β), solution treated, over aged (SOA) Ti–6Al–4V alloy (with chemical composition Ti–6.30Al–4.19V–0.19Fe–0.19O, %wt.) [83] - a material extensively employed for low temperature compressor applications, both in military and commercial aircraft jet engines. FOD-exacerbated HCF was identified in Section 2.6.2 as the key driver of military jet engine failures and associated costs – based on Section 1.2.2 direct costs with 10x multiplier (assuming direct to indirect costs multiplier for military aircraft subjected to FOD is similar to that of their commercial brethren) the annual costs incurred by military aircraft of the western air forces would be estimated at $14.86 billion (in 2013 U.S. dollars). A compressor blade specific annual cost estimate can be had from the United States Air Force's HCF Science and Technology Program – a report from the program estimated that the annual cost of HCF per blade to be $110,112 (in 2015 U.S. dollars) (the report estimated the yearly cost of HCF to be $400 million in 2000 U.S. dollars spread over 5,000 blades) [127, 128]. In order to understand how structural/cost detriment is being produced on blades of the material considered in this thesis impacted by mild steel spherical particles it is required to estimate how often those LRUs are being repaired, replaced and how such activity impacts LCCs. To estimate how often repair/replace activity occur a physical relationship must be assumed and modeled between the size of FOD particles that impact the blades, the natural growth of cracks due to tensile loading and "as-built" initial flaws on the blades – that relationship is the stuff of crack growth.

4.9.1 LEFM APPLICABILITY

Linear Elastic Fracture Mechanics (LEFM) is the branch of Solid Mechanics focused on the growth of cracks in materials that can be assumed to be isotropic and whose constitutive law is linearly elastic (e.g. Hooke's Law, where strain and stress are linearly related) [121, 122] – it is widely considered the branch focused on "large cracks", cracks whose fracture surfaces are larger than the material micro-structure in the length and width direction (the growth behavior of these types of cracks is largely 2-D) [123]. LEFM analysis is governed by the Stress Intensity Factor (denoted by K, often quoted as SIF) – a factor that universally determines the entire crack tip state of stress (it was derived from the assumption of an arbitrary crack on an arbitrary body that is arbitrarily loaded) – analysis by means of the SIF is readily supported in the literature, especially for the uniaxial loading of blades considered in this thesis [122, 126]. If the impact penetration depths calculated in Section 4.8.2.8 and from field data for reported FOD impacts on blades are larger than a threshold size based on the material microstructure LEFM analysis is valid. The deductive steps are as follows:

- A recent study determined that cracks grown from initial 0.01 inch flaws were accurately predicted by LEFM methods vs. coupon testing of Ti-6Al-4V [124]
- It is widely accepted in the Fracture Mechanics community that if the cracks are grown from initial flaws greater than 10 times the size of the material grains, LEFM is valid for analysis – for the Ti-6Al-4V alloy considered in this thesis a grain size of 25 μm is reported [125] – after multiplying by 10 that gives a threshold of 0.000250 m or 0.01 inch
- The statistical data presented in Table 4.6 for impactors of 1.33 and 3.2 mm diameter present impact penetration depths larger than the LEFM threshold

- Field data from a United States Air Force study presented in NATO's extensive report on mitigating/controlling FOD-induced HCF (mined and presented in the format of Probability Distribution Functions (PDFs) and Cumulative Density Functions (CDFs) in Figure 4-52) reveals that for reported FOD depths on compressor blades of military aircraft the vast majority were larger than the calculated LEFM threshold [17]
- Thus LEFM analysis for the cracks that will be grown from the material and loading being considered in this thesis is valid

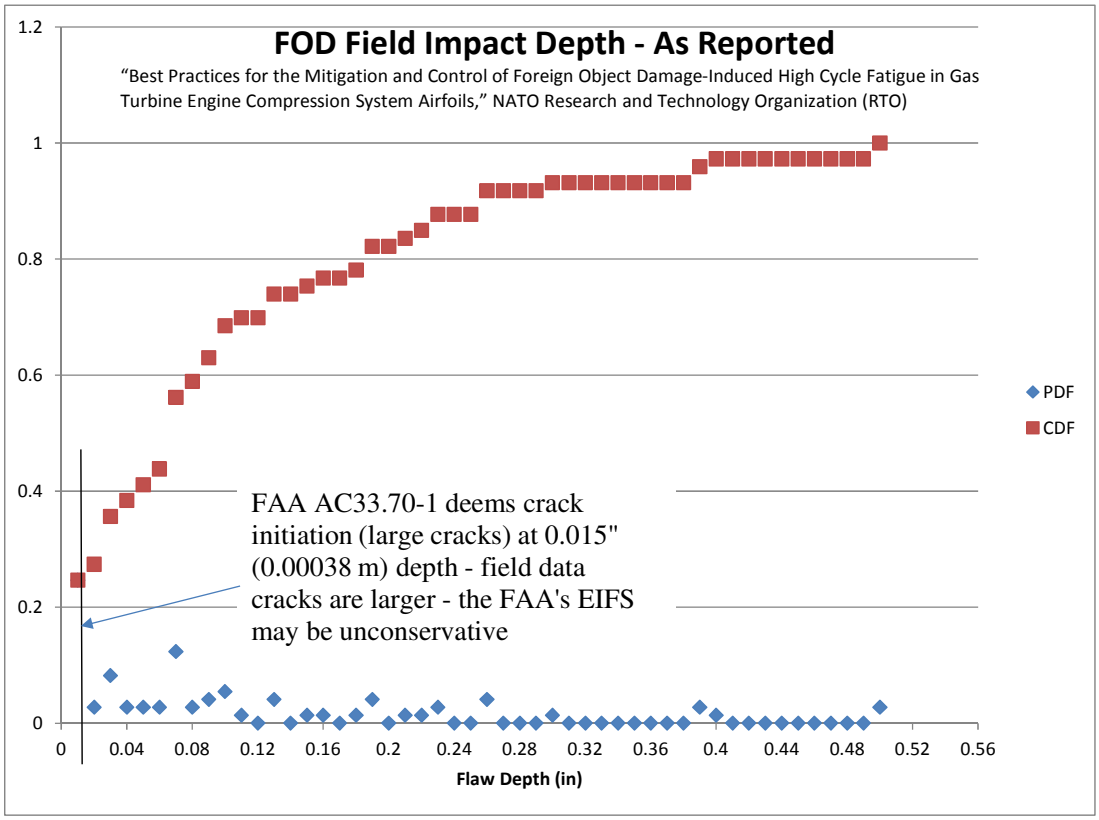


Figure 4.52: Reported FOD Depth Field Data

4.9.2 LEFM Crack Growth Theory & Application

The validity of LEFM analysis for crack growth from the material and loading considered in this thesis has been confirmed in the previous section. A discussion is now

required to substantiate the formulae for characterizing/determining, assessing and calculating the state of stress at the crack tip, the geometry factors that influence that state of stress and the cycle-by-cycle growth of the associated crack respectively.

4.9.2.1 Stress Intensity (K)

In the case of LEFM analysis, the Stress Intensity K (in units of stress$\cdot\sqrt{\text{Length}}$, Pa$\cdot\sqrt{\text{m}}$ or psi$\cdot\sqrt{\text{in}}$) not to be confused with a stress riser referred to as a Stress Concentration k_t, is a factor that fully determines the state of stress at the tip of the crack being grown [126] – often in the lingo of Fracture Mechanics K is also referred to as a Stress Intensity Factor (SIF). The origins of K are inductive in nature:

- For an arbitrary body, arbitrarily loaded in Mode I (commonly referred to as "opening mode"), with an arbitrary crack and no constraints on out-of-the-page contraction the resulting stress distribution at and away from the crack tip are described by the left hand side of Figure 4-53 – anywhere on a plane of the body the right hand side of Figure 4-53 and Equations 183 - 186 describe the stress state at a point
- For a Mode I loaded crack split evenly by and along a plane ($\theta = 0°$) and whose tip is at $r = 0$ (it can be shown that this provides the lowest energy path to grow a crack in the configuration considered in this thesis), the stress state at points along the crack surface is described by Figure 4-54 and Equations 187 - 189 – the σ_{xx} normal stresses and τ_{xy} shear stresses are automatically zero as at the free surface (the crack tip in this case) as they are unopposed – sufficiently away along x from the crack tip σ_{xx} stress is zero while the σ_{yy} stress becomes the reference stress σ (stress resulting from Mode I load)

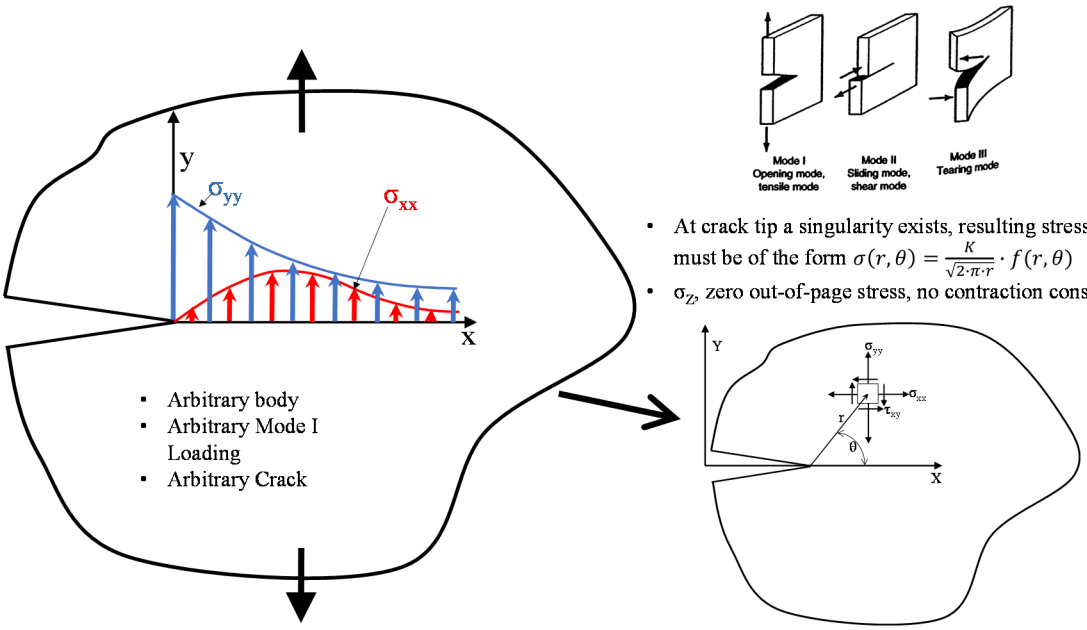


Figure 4.53: Stress Field along Crack Surface, Point Stress State

$$\sigma_{xx} = \frac{K}{\sqrt{2\cdot\pi\cdot r}}\cdot\cos\frac{\theta}{2}\cdot\left(1-\sin\frac{\theta}{2}\cdot\sin\frac{3\theta}{2}\right) \tag{183}$$

$$\sigma_{yy} = \frac{K}{\sqrt{2\cdot\pi\cdot r}}\cdot\cos\frac{\theta}{2}\cdot\left(1+\sin\frac{\theta}{2}\cdot\sin\frac{3\theta}{2}\right) \tag{184}$$

$$\tau_{xy} = \frac{K}{\sqrt{2\cdot\pi\cdot r}}\cdot\cos\frac{\theta}{2}\cdot\sin\frac{\theta}{2}\cdot\cos\frac{3\theta}{2} \tag{185}$$

$$\sigma_z = 0 \tag{186}$$

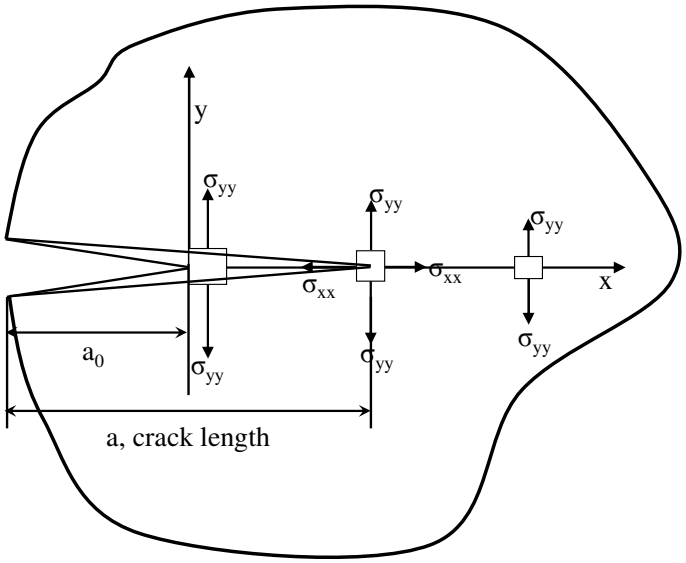


Figure 4.54: Stress State at Points along Crack Surface

$$\sigma_{xx} = \frac{K}{\sqrt{2 \cdot \pi \cdot x}} \tag{187}$$

$$\sigma_{yy} = \frac{K}{\sqrt{2 \cdot \pi \cdot x}} \tag{188}$$

$$\tau_{xy} = 0 \tag{189}$$

- The crack opening σ_{yy} stress must be proportional to the reference stress – Equation provides the proportionality, to make it an equality a constant B is added – after replacing B by $\beta \cdot \sqrt{\pi}$ and from equating Equations 189 and 191 the commonly known form of Stress Intensity is described by Equation 192 – henceforth the β are called Geometry Factors as they relate and quantify the effect of geometry around the crack tip to the overall Stress Intensity

$$\sigma_{yy} \propto \frac{B \cdot \sigma \cdot \sqrt{a}}{\sqrt{2 \cdot \pi \cdot x}} \tag{190}$$

$$\sigma_{yy} = \frac{\beta \cdot \sqrt{\pi} \cdot \sigma \cdot \sqrt{a}}{\sqrt{2 \cdot \pi \cdot x}} \tag{191}$$

$$K = \beta \cdot \sigma \cdot \sqrt{\pi \cdot a} \tag{192}$$

The stress intensity value at which fracture occurs is called the Fracture Toughness – K_c for Plane Stress or K_{Ic} for Plane Strain Fracture Toughness. Recall that the Planes Stress condition arises when through-the-thickness strains in a body are not opposed, thus no an opposing out-of-plane stress does not exist, only stresses in the plane of the body – this is generally true in slender bodies where free through-the-thickness expansion or contraction is allowed (in elastic materials like metals, where the Poisson ratio ν is positive). The Plane Strain condition arises when expansion or contraction is constrained – the means of constrain are out-of-plane stresses that allow no out-of-plane strains – typically thicker bodies encounter this condition. In the context of this thesis the test to be performed is whether the Plane Stress or Plane Strain condition results from the thickness

of the largest chord station on a fan blade – specifically the maximum thickness of the airfoil is 10% of chord, giving a maximum thickness of 0.0079 m (0.10·0.079 m), the test follows the generally accepted Fracture Mechanics test for Plane Strain [126]:

$$Material\ Thickness = t > 2.5 \cdot \left(\frac{K}{F_{ty}}\right)^2 \tag{193}$$

For t = 0.0079 m, K = 147 MPa·$\sqrt{\text{m}}$ and the Yield Strength F_{ty} = 930 MPa for Ti-6Al-4V [132] Plane Stress is confirmed (K should be referred to as K_c from this calculation),

$$0.0079\ m > 2.5 \cdot \left(\frac{147\ \text{MPa} \cdot \sqrt{\text{m}}}{930\ \text{MPa}}\right)^2 = 0.062\ m$$

Where for a given thickness t a value of K (plot provides K in units of Msi·$\sqrt{\text{in}}$, this thesis considers only SI units, thus units for K are converted toMPa · $\sqrt{\text{m}}$) is extracted from a K vs. thickness plot for the material of the blade (see Figure 4-55),

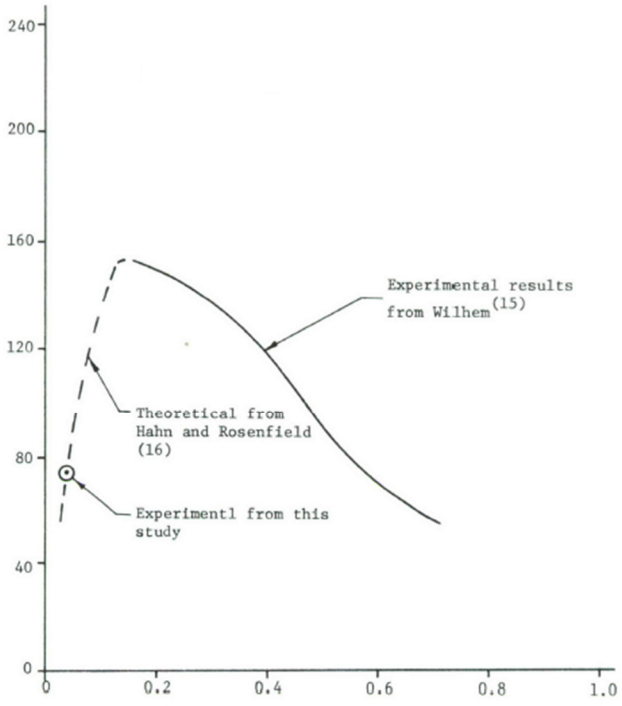


Figure 37 K_c vs. thickness for 6Al-4V titanium material.

Cox, D., Tetelman, A.S., "Improved Fracture Toughness of Ti-6Al-4V Through Controlled Diffusion Bonding", Technical Report AFML-TR-71-264, February 1972.

Figure 4.55: Fracture Toughness vs. Thickness

Values of Fracture Toughness were extracted from Figure 4-55 and a polynomial model fit was performed as this key material property is needed in the extraction of critical crack sizes that are the triggers of replacement activity of the LRUs. Measures of goodness of fit (five figures of merit must be checked to ensure goodness of fit: R^2 values greater than 0.90, actual by predicted plots whose data points are randomly scattered along a 45 degree line, residual by predicted plots whose data points present a random scattering with magnitudes much smaller compared to the predicted values, Model Fit Error (MFE) of the model relative to actual values at points employed to make the model and Model Representation Error (MRE) of the model relative to actual values at points not employed to make the model) and the fit were created with SAS Statistical Discovery Software JMP® and are presented in Figure 4-56 and Equation 194 respectively.

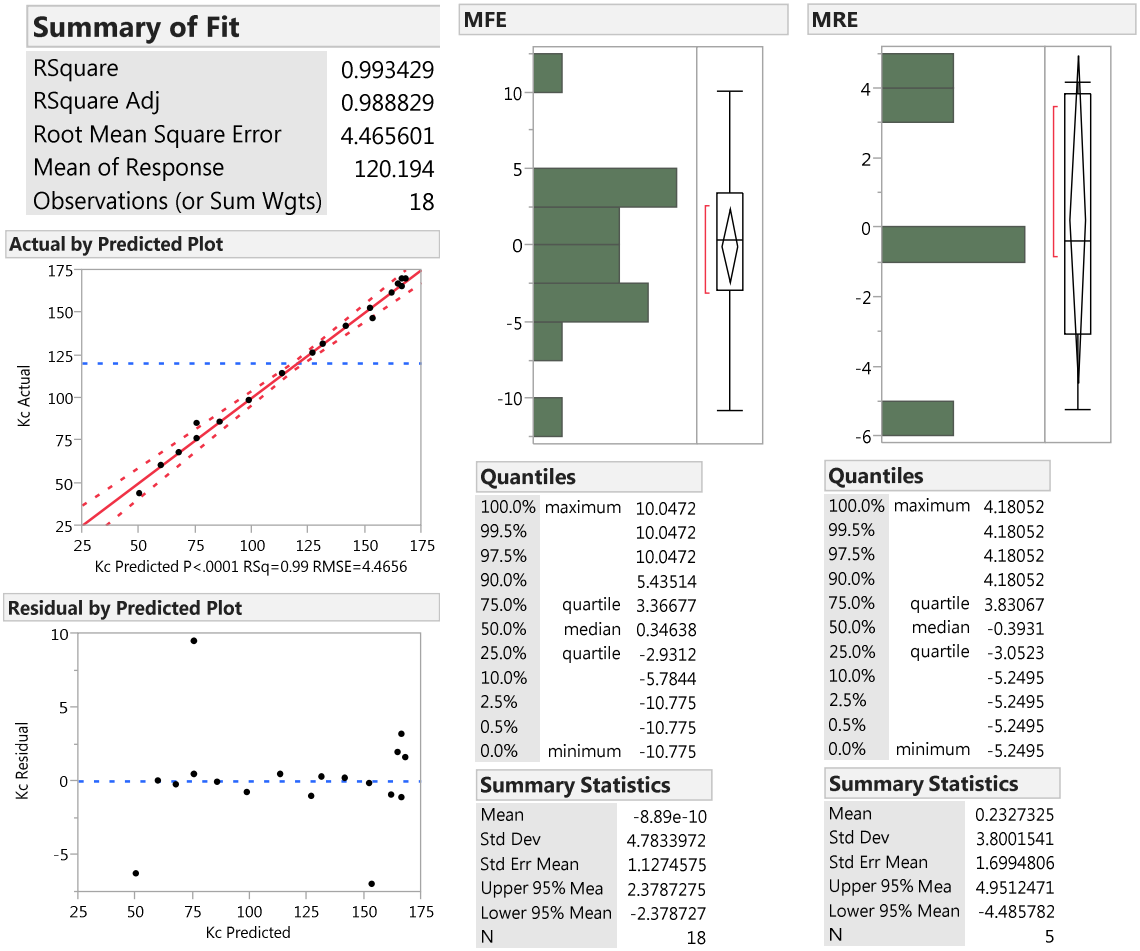


Figure 4.56: Fracture Toughness Goodness of Fit Measures

$$K_c = 222.795 + -9579.982 \cdot t_{max,airfoil} + (t_{max,airfoil} - 0.0079) \cdot ((t_{max,airfoil} - 0.0079) \cdot 164325.6553) + (t_{max,airfoil} - 0.0079) \cdot ((t_{max,airfoil} - 0.0079) \cdot ((t_{max,airfoil} - 0.0079) \cdot 8.9110 \cdot 10^7)) + (t_{max,airfoil} - 0.0079) \cdot ((t_{max,airfoil} - 0.0079) \cdot ((t_{max,airfoil} - 0.0079) \cdot ((t_{max,airfoil} - 0.0079) \cdot -3.457907983310769 \cdot 10^{10}))) + (t_{max,airfoil} - 0.0079) \cdot ((t_{max,airfoil} - 0.0079) \cdot ((t_{max,airfoil} - 0.0079) \cdot ((t_{max,airfoil} - 0.0079) \cdot ((t_{max,airfoil} - 0.0079) \cdot 2.5231 \cdot 10^{12})))) \quad (194)$$

4.9.2.2 Geometry Factors (β's)

The previous section detailed how Stress Intensity was derived from a generalized loading, crack on an arbitrary body. Inherent in Equation 192 is the effect of Geometry Factors on the overall magnitude of the SIF – for a given crack length a there is a linear

relationship between Geometry Factors (βs) and K. So what do βs account for? How are they calculated? How are multiple geometric effects accounted for by βs?

Geometry Factors can account for, but are not limited to [126]:

- Finite geometry – loaded body width, thickness, length etc.
- Shape of the crack front – crack fronts can be assumed to take several shapes including, but not limited to circular, elliptical, straight – often simple shapes approximate laboratory observations of crack shape with high accuracies, e.g.
- a Part Thru crack models the growth of a crack from the edge of a slender plate accurately
- Effect of Free Surfaces – Free Surfaces where the crack begins to grow (called Back Free Surfaces) and ahead of the crack front (called Front Free Surfaces) have empirically derived (from tables that contain test data) effects

βs are calculated by:

- Use of empirical solutions
- Finite Elem Modeling (FEM) approximations

Multiple geometric features captured by βs can be combined for a collective effect:

- Compounding
- Superposition
- Engineering judgement

4.9.2.2.1 βs, Airfoil Mid-Section

The mid-section of the NACA 65-210 airfoil is thick compared to the trailing and leading edge (the airfoil mid-section for this thesis spans from 5% to 80% chord). When a FOD particle impacts the mid-section it leaves a footprint that bears it's geometry up to the

depth of penetration on the blade (see Figure 4-57) – it is assumed that the impact site geometry produces cracks of the Thumbnail type – this assumption is supported by evidence in the literature (see Figure 4-58) [130]. The growth of the crack from its initial size ("as-manufactured" Equivalent Initial Flaw Size, EIFS) or from the impact flaw geometry bears the same compounded formulation for βs presented in Equation 193 and Figure 4-59. Superposition was not required for uniaxial loading.

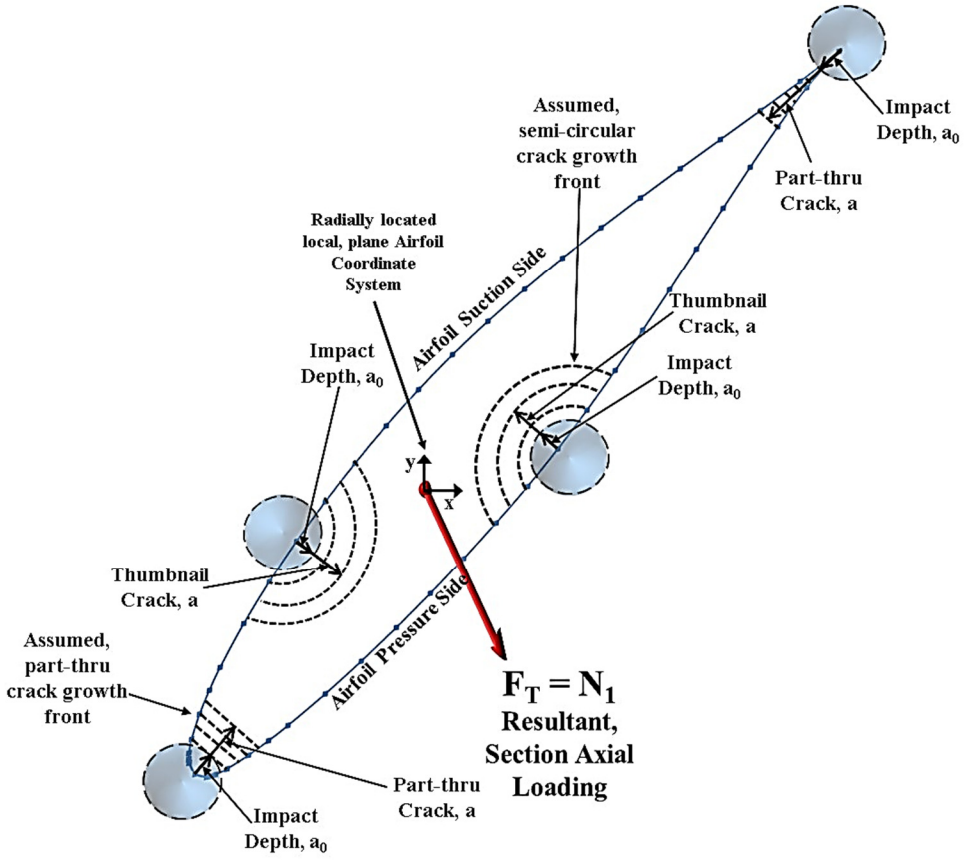


Figure 4.57: Airfoil Sample Impact Locations & Assumed Crack Profiles

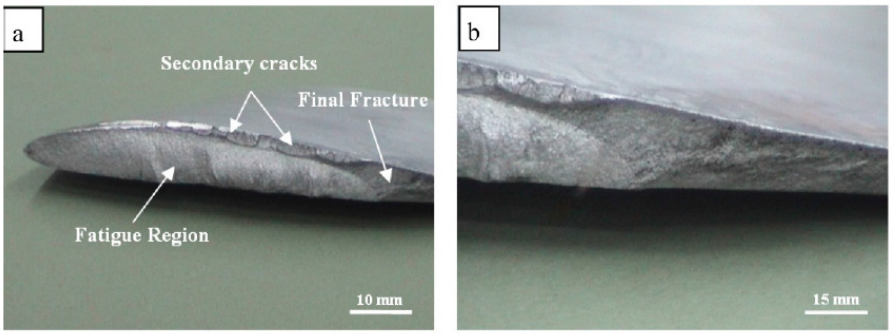


Figure 7. (a) Fracture surface of blade No. 8, (b) transition from fatigue to tensile overload fracture

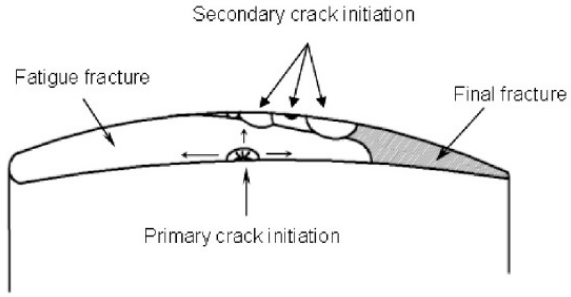


Figure 4.58: Compressor Airfoil Crack Growth Geometry

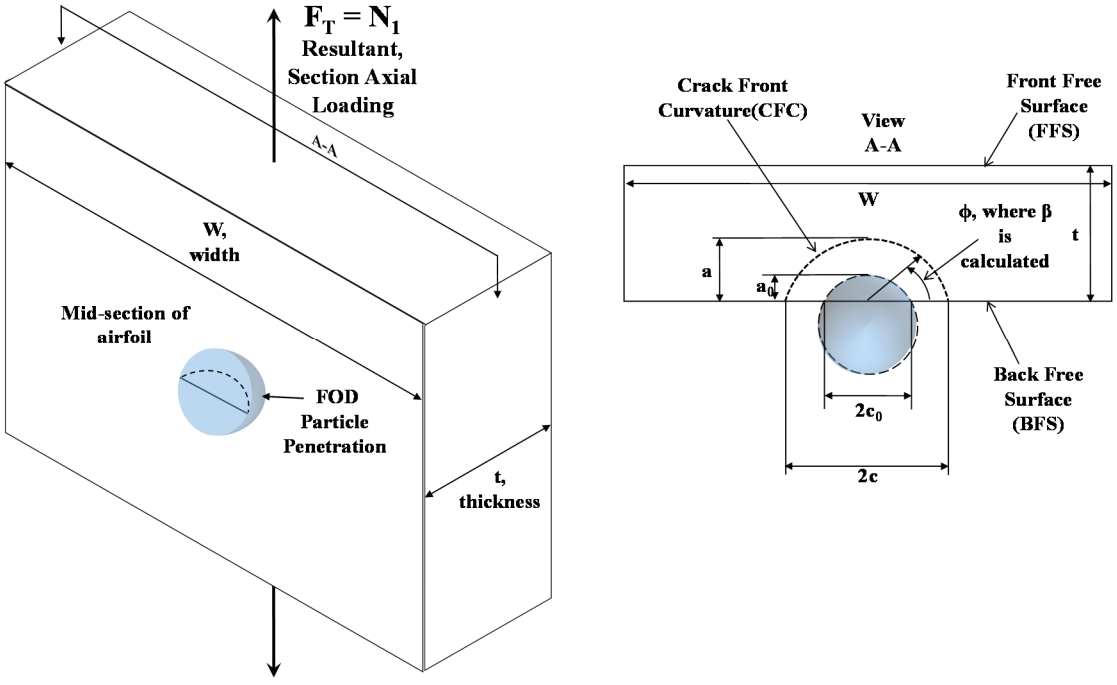


Figure 4.59: Impacted Airfoil Mid-Section βs

$$K = \beta_{BFS} \cdot \beta_{FFS} \cdot \beta_W \cdot \beta_{CFC} \cdot \sigma \cdot \sqrt{\pi \cdot a} \tag{195}$$

Where the compounded βs are calculated as follows:

- The Back Free Surface β, $\beta_{BFS} \cong 1.12$
- The Front Free Surface β is estimated from an empirical table [126], therefore,

$$\beta_{FFS} = \beta_{FFS}\left(\frac{a}{2 \cdot c}, \frac{a}{t}\right) \tag{196}$$

A Response Surface Fit of the empirical table reported by Broek was performed to sufficient goodness [126] – Figure 4-60 presents the goodness of fit measures and Equation 197 the fit.

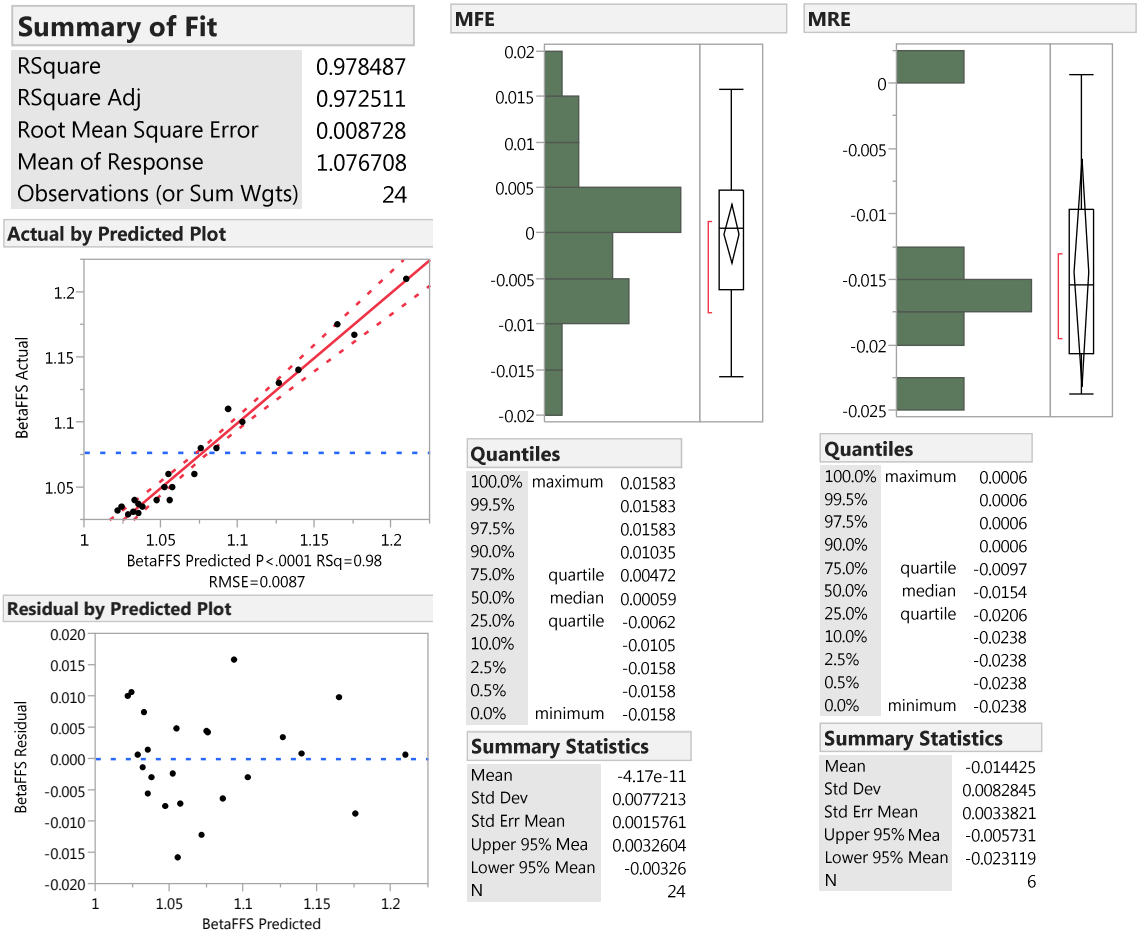


Figure 4.60: β_{FFS} Goodness of Fit Measures

$$\beta_{FFS} = 1.0888 + 0.1911 \cdot (a/t) + -0.2275 \cdot (a/2c) + ((a/t) - 0.2375) \cdot (((a/t) - 0.2375) \cdot -0.33625) + ((a/t) - 0.2375) \cdot (((a/2c) - 0.2567) \cdot -0.6423) + ((a/2c) - 0.2567) \cdot (((a/2c) - 0.2567) \cdot 0.4435) \tag{197}$$

- The width of the airfoil mid-section is finite and the location of the impact site has eccentricity from the center of the airfoil mid-section (impact sites may vary), a β must account for both geometric features – a force balance method derived β_W that accounts for finite width and eccentricity is as follows [129]:

$$\beta_W = \frac{\pi \cdot G_4}{G_1 \cdot G_4 + G_2 \cdot G_3} \quad (198)$$

$$e = ChordatImpact \cdot Abs[XpercentChord - 0.5] \quad (199)$$

$$G_1 = \cos^{-1}[2c/(ChordatImpact - 2 \cdot e)] \quad (200)$$

$$G_2 = \cos^{-1}[2c/(ChordatImpact + 2 \cdot e)] \quad (201)$$

$$G_3 = \ln[(ChordatImpact/2 - e + ((ChordatImpact/2 - e)^2 - (2c/2)^2)^{0.5})/(2c/2)] \quad (202)$$

$$G_4 = \ln[(ChordatImpact/2 + e + ((ChordatImpact/2 - e)^2 - (2c/2)^2)^{0.5})/(2c/2)] \quad (203)$$

- The Crack Front Curvature is assumed to conform to the geometry of the assumed infinitely stiff impactor (the FOD particle) per Figure 4-46 and Equation 181, conservatively the β is calculated at $\phi = 0$ – Broek's solution for β_{CFC} is as follows [126]:

$$\beta_{CFC} = \frac{(a/c)^{0.5}}{\int_0^{\pi/2} \{1-(1-(a/c)^2) \cdot (\sin\phi)^2\} d\phi} \quad (204)$$

Where,

t – local thickness at impact site

2c – local width at impact site, estimated from Equation 181

4.9.2.2.2 βs, Airfoil Leading and Trailing Edge

For the leading and trailing edge of the NACA 65-210 airfoil cracks are assumed to be of the Part-Thru type because the localized thickness much smaller than chord of the airfoil. The formulation for Stress Intensity and β are as follows [126]:

$$K = \beta_W \cdot \sigma \cdot \sqrt{\pi \cdot a} \tag{207}$$

$$\beta_W = 1.12 - 0.231 \cdot (\text{aa/ChordatImpact}) + 10.55 \cdot (\text{aa/ChordatImpact})\text{^}2 - 21.71 * (\text{aa/ChordatImpact})\text{^}3 + 30.382 * (\text{aa/ChordatImpact})\text{^}4 \tag{208}$$

4.9.2.3 Crack Growth Rate (da/dN)

Crack Growth Rate (da/dN) is the amount a crack a will grow in one cycle N. Crack Growth Rate is a function of the Stress Intensity range ΔK (K_{max}-K_{min}) and Load Ratio R (Equation 209) – as ΔK or R increase so does the rate of crack growth – R is the ratio of minimum stress to maximum stress in one cycle ($\sigma_{min}/\sigma_{max}$). The integration of da/dN in cycles or crack size increments yields crack size vs. loading cycles curves (see Figure 4-61) that are the basis of fracture control – the slope and magnitude of the curves are employed to determine if a component has to be repaired at a given cycle or if its structural life is compromised.

$$\frac{da}{dN} = f(\Delta K, R) \tag{209}$$

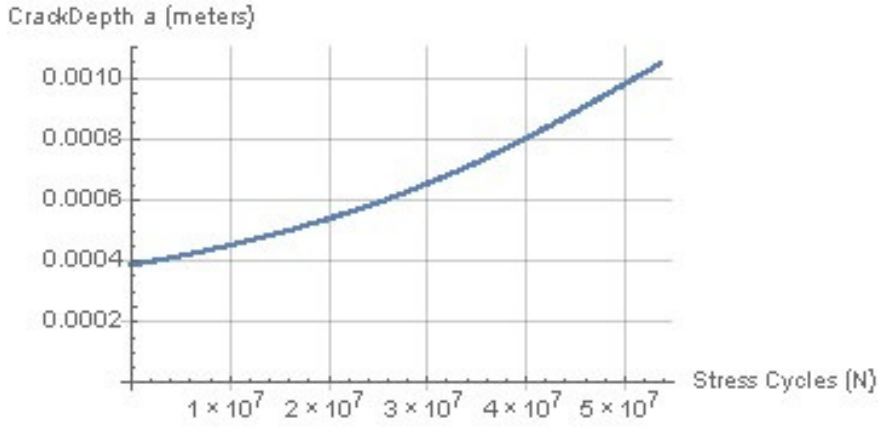


Figure 4.61: Sample Crack Growth Plot

Recall from the formulation for Stress Intensity (Equation 192), that K is related to a geometry factor β and the reference stress σ linearly ($K = \beta \cdot \sigma \cdot \sqrt{\pi \cdot a}$) – as this relationship represents the Stress Intensity at a crack size a and the reference stress σ is independent of crack size (recall from Section 4.8.2.5 that the reference stress is primarily a function of angular velocity and radial location) ΔK is a related to the reference stress by Equation 210. The reference stress σ is related to the load ratio R by means of Equations 212 and 213.

$$\Delta K = \beta \cdot \Delta\sigma \cdot \sqrt{\pi \cdot a} \tag{210}$$

$$\Delta\sigma = \sigma_{max} - \sigma_{min} \tag{211}$$

$$R = \frac{\sigma_{min}}{\sigma_{max}} \tag{212}$$

$$\sigma = \sigma_{mean} = \frac{\sigma_{max}+\sigma_{min}}{2} = \frac{\sigma_{max}\cdot(1+R)}{2} \tag{213}$$

Equations 208 and 209 yield,

$$\sigma_{max} = \frac{2\cdot\sigma}{1+R} \tag{214}$$

$$\sigma_{min} = \frac{2\cdot R\cdot\sigma}{1+R} \tag{215}$$

With an understanding and formulae for ΔK, R and Δσ the question of the form of the rate equation can now be addressed. Formulation for da/dN is test specific – curves are fit through test data for da/dN vs. ΔK points at a load ratio R – this empirical relationship, embodied in Equation 216, is known in the field of Fracture Mechanics as the Paris law or equation [126]. For the material considered in this thesis, Ti-6Al-4V, a modified version of the Paris law was employed by Ritchie, et al to account for tests conducted at varying load ratios [131] – that specific formula for da/dN (Equation 217) is pivotal to growing cracks from the loading configuration and material (the material considered by Ritchie, et

al is the same Ti-6Al-4V alloy) considered in this thesis as the residual, compressive stresses from beneficial surface treatments such as Laser Shock Peening (LSP) modify the effective load ratio – the assumption is made that the residual, compressive stresses from surface treatments can be super-posed on the reference stresses ("Centrifugal" stresses on an airfoil section resulting from blade rotation) on the airfoil section. The formula provides a powerful capability - it was built from data from tests on large cracks (>5mm) whose worst-case lower-bound ΔKs ($\Delta K_{threshold}$) are able to describe the early growth of smaller cracks emanating from simulated FOD impact sites - the assumption is made that the formula captures the growth rate trends in the Ti-6Al-4V alloy blades considered in this thesis with acceptable accuracy. Figure 4-62, taken from Ritchie et al, strengthens the case to use their formulation to grow cracks under the conditions considered in this thesis [131] – recall from Equation 210 that ΔKs are affected by βs directly, FOD particle impact geometries if more severe than naturally growing cracks account for the jump in crack growth rates evinced in the figure.

$$\frac{da}{dN} = C_p \cdot \Delta K^{m_p} \tag{216}$$

$$\frac{da}{dN} = 5.2 \cdot 10^{-12} \cdot \Delta K^{2.5} \cdot {K_{max}}^{0.67} \tag{217}$$

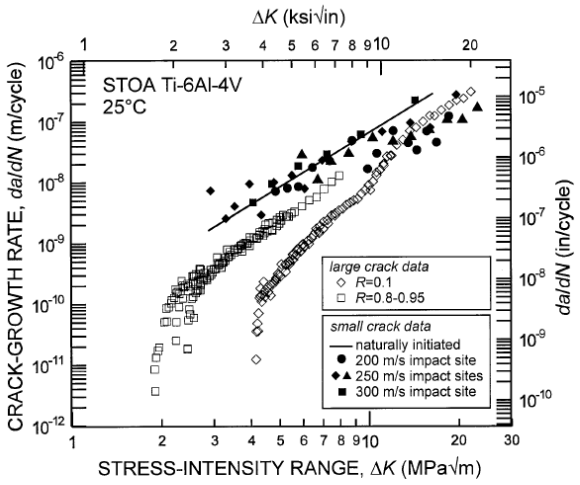


Fig. 7 Fatigue-crack propagation results for naturally initiated small (~45–1000 μm) surface cracks at $R = 0.1$ (line)[24] and for small cracks emanating from a variety of FOD impact sites (closed symbols) in the solution-treated and overaged Ti–6Al–4V alloy, as compared to through-thickness, large cracks (>5 mm) at $R = 0.1$ and 0.8–0.95. All measurements were made in room air.

Ritchie, R.O., Davidson, D.L., Boyce, B.L., Campbell, J.P., Roder, O., "High-cycle Fatigue of Ti–6Al–4V", 1999.

Figure 4.62: da/dN vs. ΔK Curve for Ti-6Al-4V

Implicit in the crack growth rate formulation is a significant assumption whose consequences will be made explicit: loading on the airfoil is of constant amplitude and high R (in this thesis R larger or equal to 0.8, to leverage public domain impact test data [92, 118, 131]), as such there is no retardation – growth is not retarded due to negligible plasticity effects at the crack tip.

4.9.2.4 Residual Compressive Stress Treatments

As was discussed in the previous sections crack growth is influenced by loading and geometry of the crack and loaded structure (review Equations 209 and 210). If a wholesale design change of the blade geometry is not possible and the size/geometry of the FOD particles that impact the blade or of the initial, “as-built” cracks cannot be changed a less-invasive beneficial procedure needs to be considered to reduce the effective reference stress that along with Geometry Factors drives Stress Intensity and by extension the Crack Growth Rate (da/dN). An additional consideration that de-incentivizes geometric changes

stems from the way compressor blades are designed – the current design paradigm prioritizes performance, blades are optimized geometrically for efficiency – the engine OEMs may not be keen on having their ultra-efficient designs' Outer Mold Lines (OMLs) changed.

4.9.2.4.1 Comparison of Dominant Treatments

There are three dominant treatments that infuse beneficial compressive stresses from the surface to a depth on a compressor blade: Shot Peening, Low Plasticity Burnishing and Laser Shock Peening. These three surface treatments work on the same principle: a residual, compressive stress gradient is imparted that to an effective penetration depth that effectively reduces reference stress (by extension reducing the Load Ratio R) and thus slows down the rate of crack growth [132, 133, 134, 135].

Shot Peening is the oldest residual, compressive stress surface treatment – the treatment provides a limited compressive stress gradient from the surface of the material treated with significant surface roughness – because the surface roughness effectively changes the geometry of the material treated and it's limited treatment penetration depth it will not be considered for the compressor blades in this thesis (see Figure 4-63) [135, 136]. Low Plasticity Burnishing (LPB) employs a smooth, free rolling spherical ball to impart deep compressive stresses on the material being treated with improved surface finish (see Figure 4-63)[136]. LPB has drawbacks: it can distort the surface being treated and it is has to come in contact evenly, a significant constraint for highly twisted blades. LPB will not be considered for the blades in this thesis. Laser Shock Peening (LSP) is the only non-invasive treatment that imparts deep compressive stresses with improved surface finish (see Figure 4-63) [132, 133, 134, 135]. LSP employs a high power laser to focus a short

energy pulse on a coating (typically paint) that is applied to the material being treated. The coating is constrained by a thin film of water. As the laser passes through the water it rapidly vaporizes the coating creating a fast energy shock wave that imparts dynamic stresses that exceed the dynamic yield stress of the material and thus locally, plastically deform the surface of the part/component being treated. Once the laser is removed, compressive, residual stresses that grade from the surface to a penetration depth on the material being treated remain. LSP is the treatment considered for the blades in this thesis.

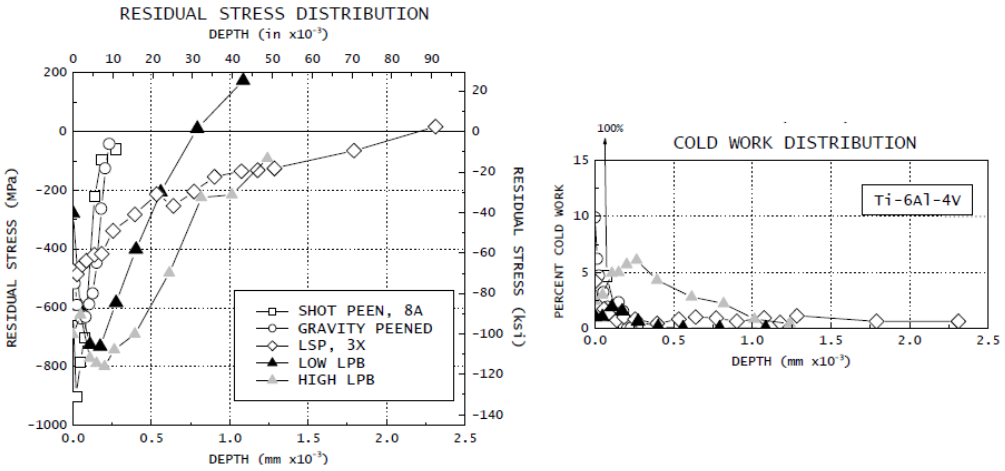


Figure 1 - Subsurface residual stress and cold work distributions produced by shot peeing (8A, 200%), gravity peening, LSP (3X), and two levels of LPB in Ti-6Al-4V.

Prevey, P.S., Shepard, M.J., Smith, P.R., "The Effect of Low Plasticity Burnishing (LPB) on the HCF Performance and FOD Resistance of Ti-6Al-4V", 2001.

Figure 4.63: Shot Peening, LPB & LSP Comparison

LSP performance data for depth of penetration, residual stress magnitude and for varying intensities for Ti-6Al-4V is scarce and/or proprietary – Brajer's work provides the only source of LSP performance data in the public domain could be mined for an LSP residual stress model (see Figure 4-64)[134]. Brajer presents residual stresses for varying penetration depths and application intensity in the form of mean values with maximum and minimum Quartile values. A simple statistical approximation is assumed to create a polynomial fit for one application intensity:

- The maximum and minimum residual, compressive stresses are approximately

$$Max_{ResidualStress} = Mean_{ResidualStress} + 3 \cdot StandardDeviation \quad (218)$$

$$Min_{ResidualStress} = Mean_{ResidualStress} - 3 \cdot StandardDeviation \quad (219)$$

- The standard deviation is approximated with Equations 218 and 219

$$Max_{ResidualStress} - Min_{ResidualStress} = 6 \cdot StandardDeviation \quad (220)$$

$$StandardDeviation = \frac{Max_{ResidualStress} - Min_{ResidualStress}}{6} \quad (221)$$

- Fits are made on the extracted data for mean, maximum and residual compressive stresses for ¼ of the minimum application intensity from Figure 4-64 – ¼ the minimum intensity was chosen based on the reference stress magnitude and the thinness of the 1st HPC compressor blades – Equations 222 - 224 present the fits and Figure 4-65 goodness of fit measures
- A normal distribution is made with the fit for mean and standard deviation made with the fits for maximum and minimum residual, compressive stresses (Equations 221 and 222) – the randomly sampled inverse Cumulative Distribution Function (CDF) value is the LSP residual stress (Equation 225)(Appendix F presents the code for the simulation for Crack Growth and LCC Estimation that contains LSP treatment model) – the formulation is as follows:

$$Mean_{ResidualStress} = (-70.7332) + 117116.2634 \cdot a + (a - 0.0003) \cdot (a - 0.0003) \cdot -395474551.7585 + (a - 0.0003) \cdot (a - 0.0003) \cdot (a - 0.0003) \cdot 600608204212.839 + (a - 0.0003) \cdot (a - 0.0003) \cdot (a - 0.0003) \cdot (a - 0.0003) \cdot -313392042189632 \quad (222)$$

$$Max_{ResidualStress} = (-91.3889) + 163881.6182 \cdot a + (a - 0.0003) \cdot (a - 0.0003) \cdot -621710641.0598 + (a - 0.0003) \cdot (a - 0.0003) \cdot (a - 0.0003) \cdot 784096231132.582 + (a - 0.0003) \cdot (a - 0.0003) \cdot (a - 0.0003) \cdot (a - 0.0003) \cdot -181928735061177 \quad (223)$$

$$Min_{ResidualStress} = (-51.0323) + 73621.0918 \cdot a + (a - 0.0003) \cdot (a - 0.0003) \cdot -186084270.7815 + (a - 0.0003) \cdot (a - 0.0003) \cdot (a - 0.0003) \cdot 369136849298.813 + (a - 0.0003) \cdot (a - 0.0003) \cdot (a - 0.0003) \cdot (a - 0.0003) \cdot -331072585461178 \quad (224)$$

From mean residual stress and standard deviation (Equation 221 and 222),

$$\sigma_{Residual} = Inv.CDF[NorDist[Mean_{ResidualStress}, StandardDeviation], RandNum] \quad (225)$$

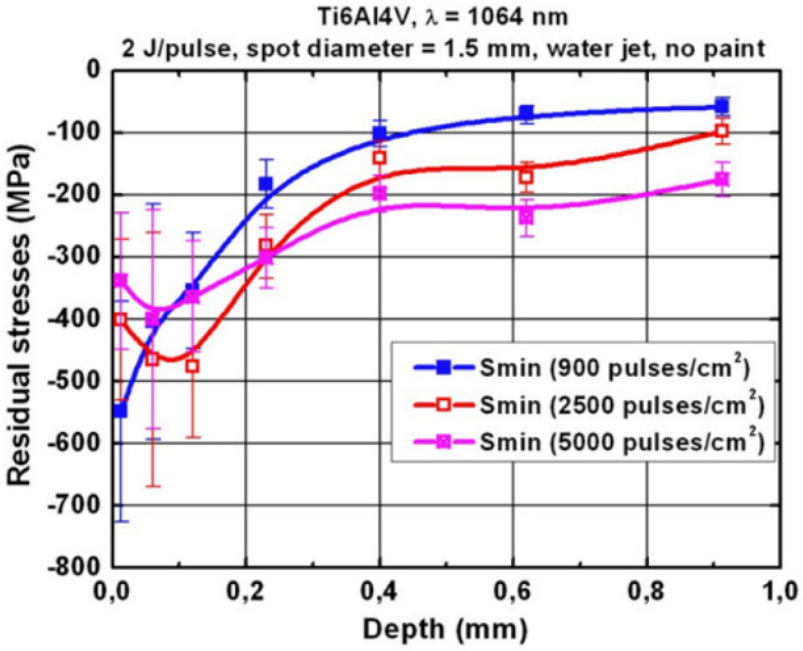
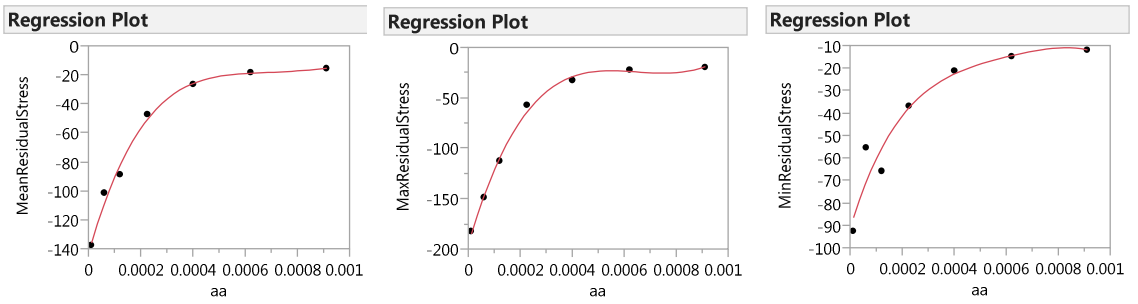


Fig. 4. *The course of residual stresses after laser shock peening processing applied on the Ti6Al4V alloy.*

Brajer, J., "Application of Laser Shock Peening", 2014.

Figure 4.64: Laser Shock Peening Gradient



Summary of Fit	
RSquare	0.991838
RSquare Adj	0.975513
Root Mean Square Error	7.447442
Mean of Response	-61.9643
Observations (or Sum Wgts)	7

Summary of Fit	
RSquare	0.996471
RSquare Adj	0.989413
Root Mean Square Error	6.782141
Mean of Response	-81.75
Observations (or Sum Wgts)	7

Summary of Fit	
RSquare	0.928999
RSquare Adj	0.786996
Root Mean Square Error	13.88919
Mean of Response	-42.4286
Observations (or Sum Wgts)	7

Figure 4.65: Goodness of Fit Measures, LSP Model

Now that the formulation has been established a question needs answering: how can the benefit of residual, compressive stresses from LSP treatments be implemented/modeled in the blades considered in this thesis? Part of the answer is in how LSP treatments are performed: high energy laser pulses are carefully overlaid to provide an averaging effect on the magnitude of residual stress that is left on the material being treated – a compressive stress gradient at the center of application (see Figure 4-64) grades from the surface to a penetration depth on the material – away from the center of application the gradient gradually reduces in compressive stress magnitude until at a distance it becomes a residual tensile stress (see Figure 4-66)[142]. The assumption is made that a suitable overlay distance is chosen to maintain the application center depth to cover the OML of the airfoil – on the extreme ends of the total, overlaid application on the airfoil residual, tensile stresses will remain (see Figure 4-67). It is precisely this residual, tensile stress that limits what treatment intensity is imparted to a component being treated – from Figure the ratio of tensile to compressive residual stresses is about 1/7, if a "large enough" compressive stress is imparted there would be 1/7 its magnitude in tensile stress at the ends

of the treatment overlay that may locally exacerbate HCF regardless of FOD impacts (by superposition this tensile stress would add to the reference tensile stress of the rotating blade). The decision was made to impart the minimum intensity gradient scaled by a factor of 4 from Figure 4-64 – it is assumed that for the FOD particles considered in this thesis the scaled gradient must also accommodate the raised stress from the impact site sizes, specifically for the of 1.33 and 3.2 mm diameter impactors the scaled gradient is of magnitude 1 and 1.5 respectively. The stress distribution on the airfoil is achieved by superposition – the Load Ratio R is affected by the addition of the residual stress to minimum and maximum section stress and becomes an effective ratio (Equation 226) while in the da/dN formulation (Equation 217) the only term that is affected is the K_{max} formulation (presented in Equation 217a). Figure 4-67 presents a physical limitation to LSP treatments – real estate has to be left on the airfoil to accommodate the residual stresses. Appendix F presents the code for simulation for Crack Growth and LCC Estimation where the reader can observe the implementation of LSP.

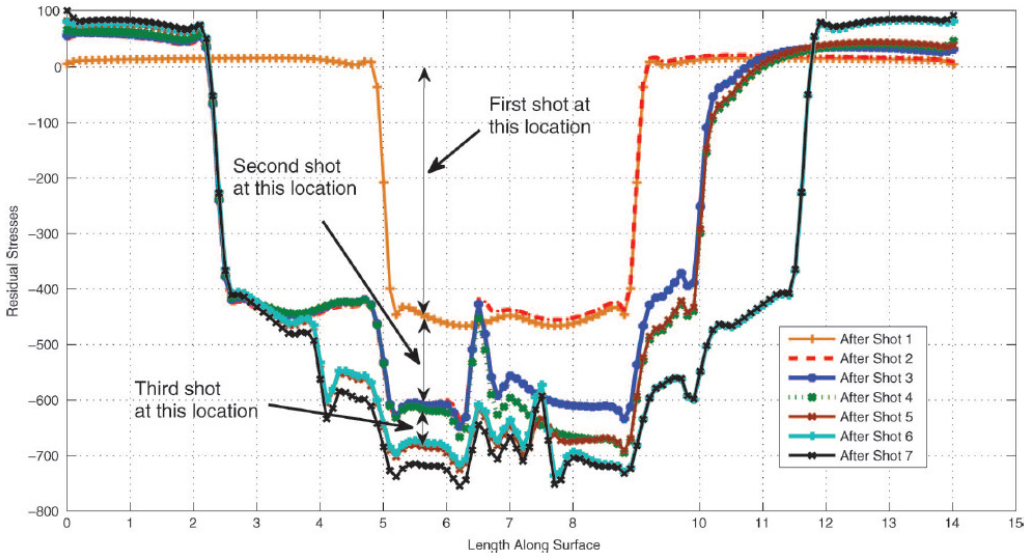


FIG. 22. Change in residual stress profile at a line on surface by seven shots. (Color figure available online)

Modeling and Parameter Design of a Laser Shock Peening Process, 2011.

Figure 4.66: LSP Residual Stress Change along Surface Treated

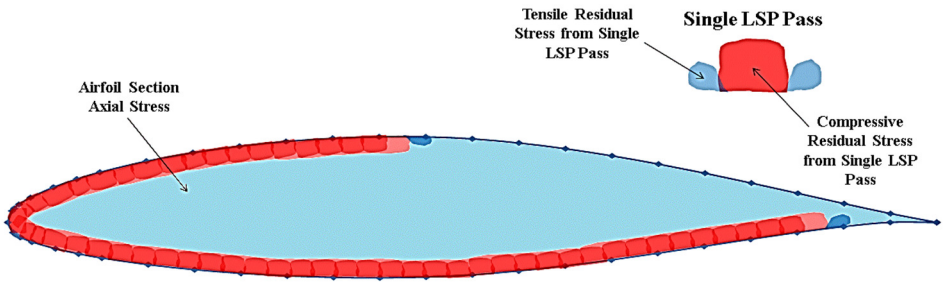


Figure 4.67: Airfoil Stress Superimposition

$$R_{effective} = \frac{\sigma_{min}+\sigma_{Residual}}{\sigma_{max}+\sigma_{Residual}} \tag{226}$$

$$\frac{da}{dN} = 5.2 \cdot 10^{-12} \cdot \Delta K^{2.5} \cdot \left[(\sigma_{max} + \sigma_{Residual}) \cdot \beta \cdot \sqrt{\pi \cdot a}\right]^{0.67} \tag{217a}$$

4.9.2.4.2 Airfoil Map for LSP Treatment

As was presented in the previous section LSP cannot be applied to the entire airfoil, "real estate" must be left on the OML of the airfoil for the residual, tensile stresses left over from overlaying LSP treatments (see Figure 4-67). Recall from Section 4.8.2 that the location where an impact occurs is known once the impact law of Section 4.8.2.6 is triggered – specifically the radius, impact side (Suction or Pressure side) and % chord from leading edge. Histogram plots of these three recorded values at impact provide a map for where LSP should be performed (see Figures 4-68 – 4-71) – the plots come from 1,081 simulations (81 Full Factorial and 1,000 Space Filling Latin Hypercube) of fan face to 1st HPC rotor FOD particle travel for FOD of 1.33 and 3.2 mm diameter. The plots reveal that impact testing only the airfoil leading edge is wrong – impact locations are distributed across the airfoil. The LSP treatment mapping that will be implemented in the simulation for Crack Growth and LCC Estimation (see Appendix F) is as follows:

- FOD particle of 1.33 mm diameter
  - On Pressure Side for 0.29 m $\leq R_{impact} \leq$ 0.39 m and $0 \leq$ % Chord ≤ 1
  - On Suction Side for 0.29 m $\leq R_{impact} \leq$ 0.39 m and $0 \leq$ % Chord ≤ 0.8

- FOD particle of 3.2 mm diameter
    - On Pressure Side for 0.30 m ≤ R_{impact} ≤ 0.39 m and 0 ≤ % Chord ≤ 1
    - On Suction Side for 0.30 m ≤ R_{impact} ≤ 0.39 m and 0 ≤ % Chord ≤ 0.39

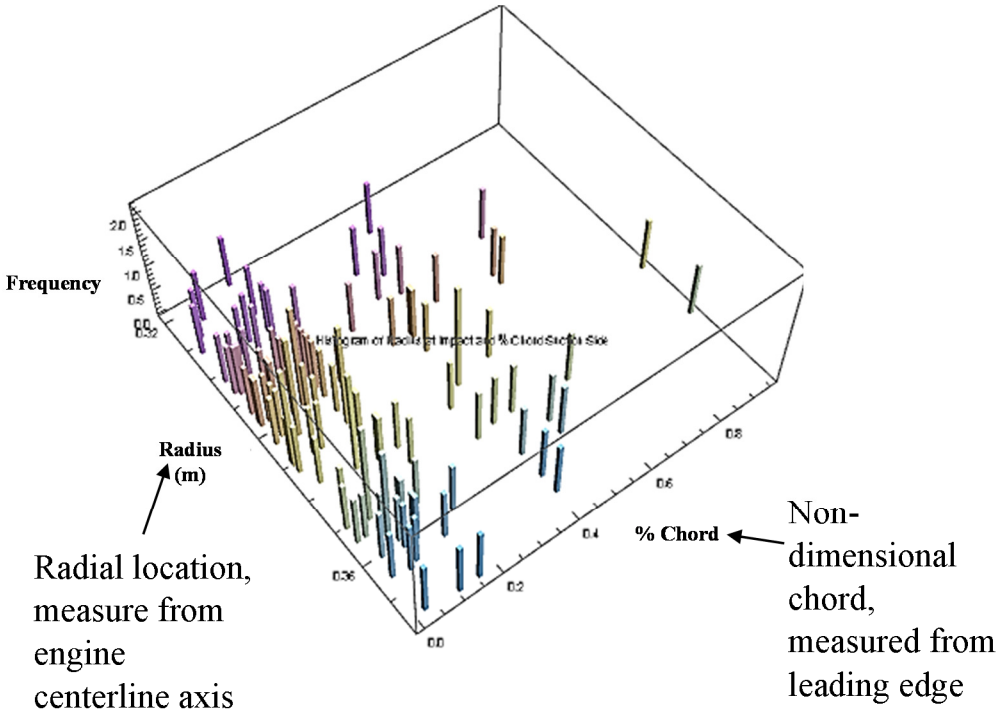


Figure 4.68: Suction Side Radius vs. % Chord Impact Locations, 1.33 mm FOD

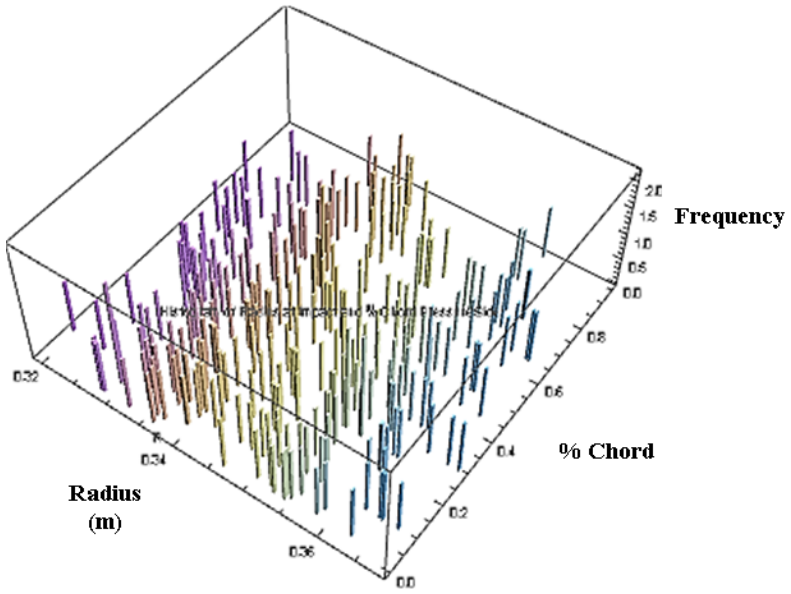


Figure 4.69: Pressure Side Radius vs. % Chord Impact Locations, 1.33 mm FOD

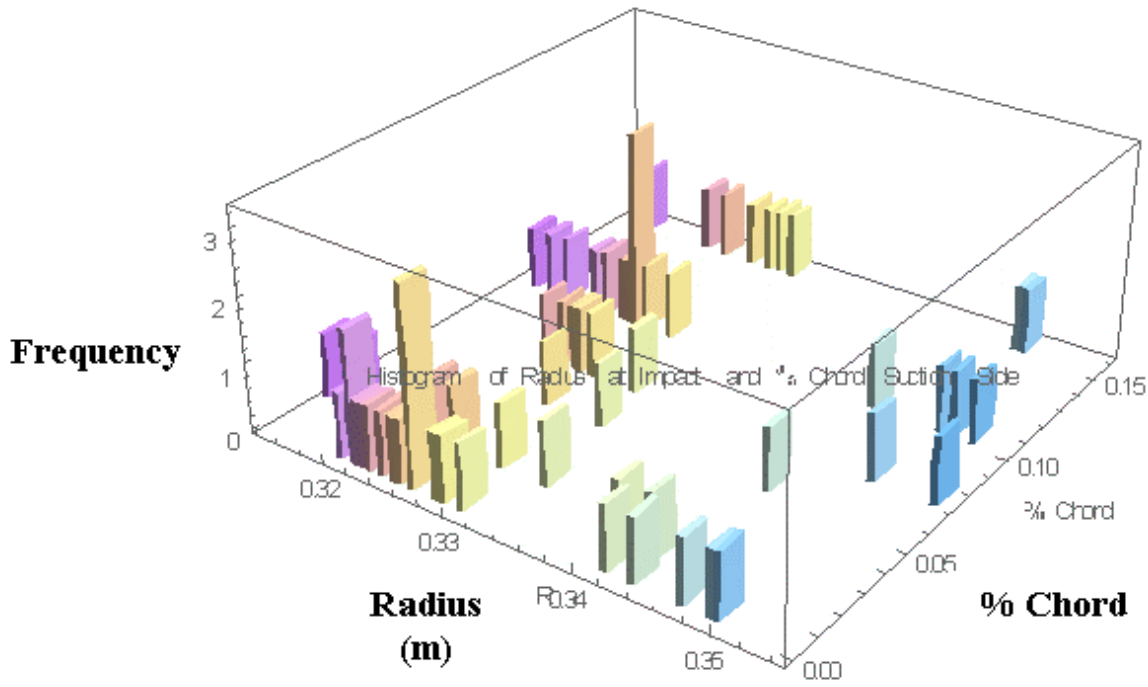


Figure 4.70: Suction Side Radius vs. % Chord Impact Locations, 3.2 mm FOD

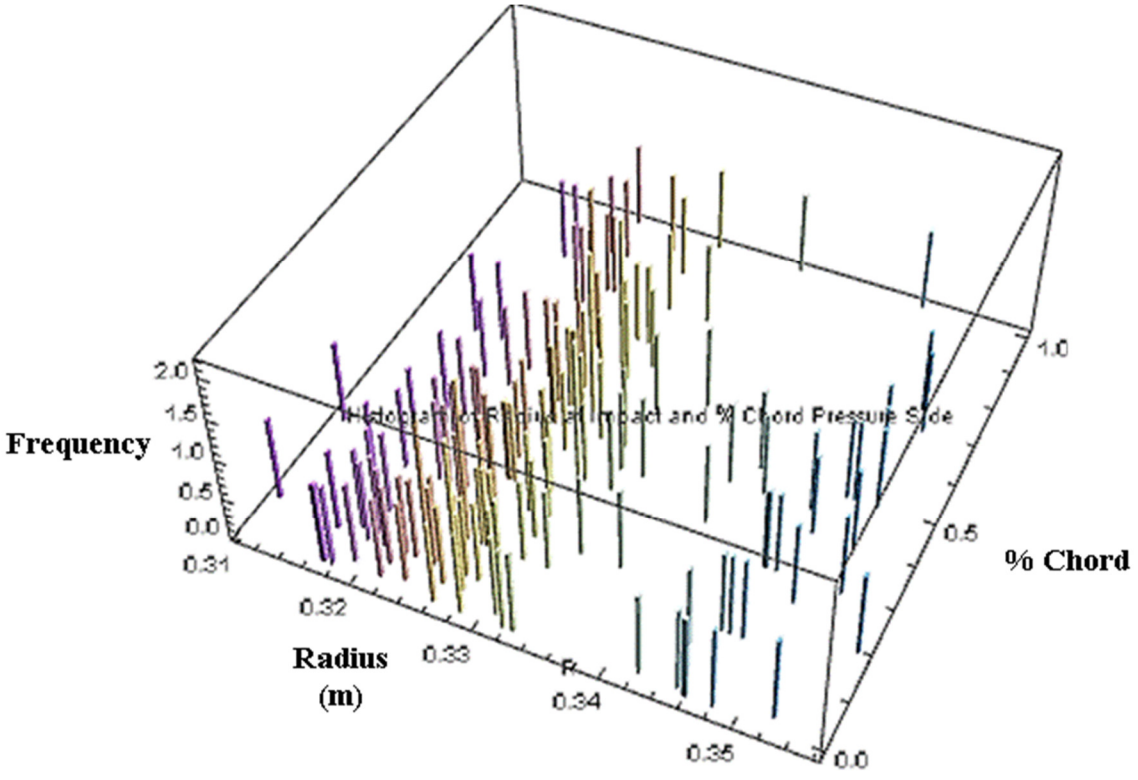


Figure 4.71: Pressure Side Radius vs. % Chord Impact Locations, 3.2 mm FOD

In addition to the locations where the impacts occur radially from the engine centerline axis and on the airfoil the simulation code for particle travel (if ingested through the fan, Appendix E) records the impact velocities. Though not considered in this thesis, future research work may consider the magnitude of impact velocities on the airfoil to determine where, locally the LSP treatment needs to have increased intensity. Figures 4-72 – 4-75 present the histograms for the chord-wise location and impact velocity on the airfoil. These plots reveal that most impact tests reported in the public domain erroneously fix impact velocity mostly at the leading edge around 300 m/s [17, 87, 88, 89, 90, 91] – this research effort reveals that impact velocity varies from impactor size, airfoil side and spool speed. On the Suction Side of the airfoil the impact velocities are overwhelmingly the function of the particle's axial velocity – they are much slower than on the Pressure Side. On the Pressure Side another key finding is revealed: it is the wheel velocity ($2 \cdot \pi \cdot R_{impact} \cdot N_2/60$) at the impact location on a blade on the 1st HPC that drives the magnitude of impact velocities – like a paddle or a racket particles are struck by the blade.

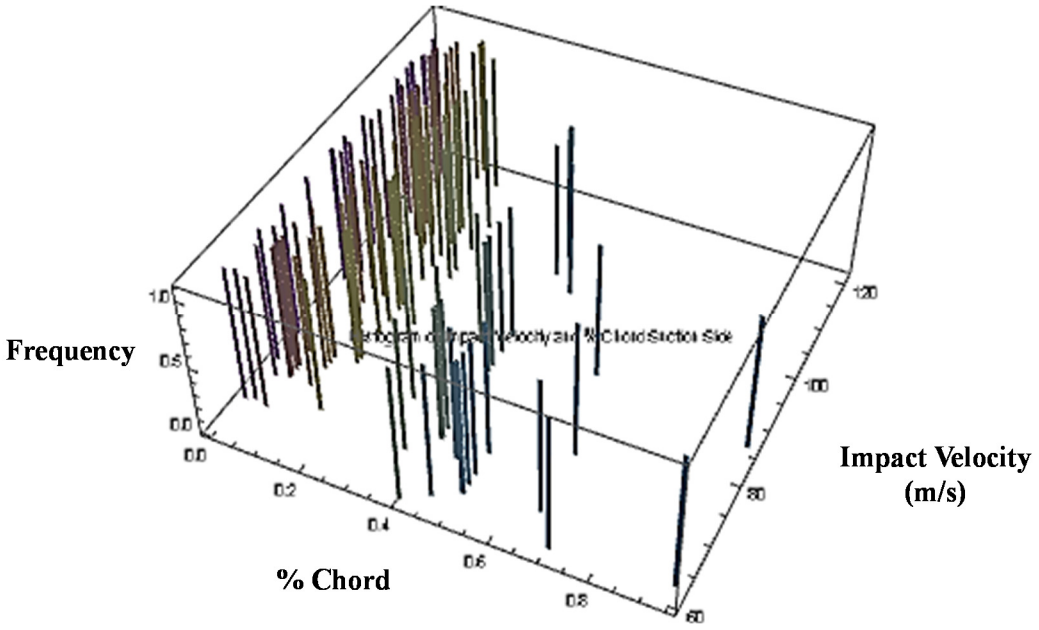


Figure 4.72: Suction Side % Chord vs. Impact Velocity, 1.33 mm FOD

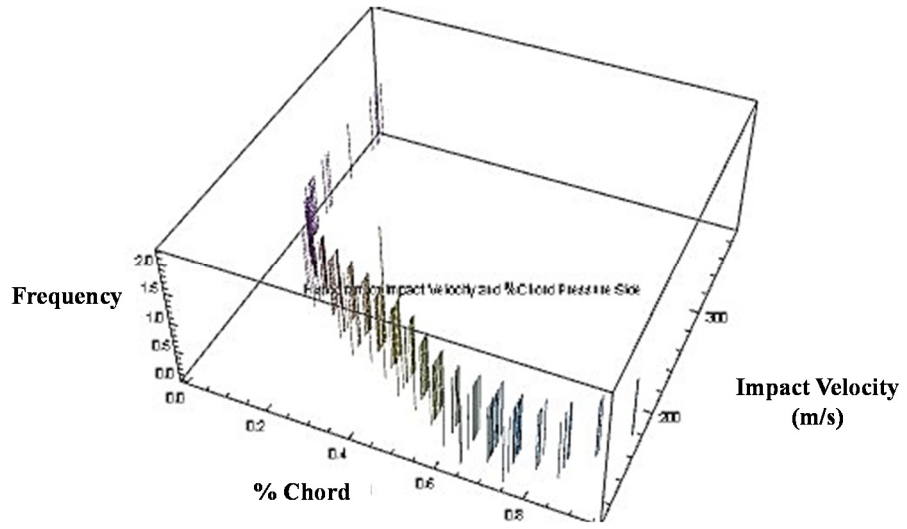


Figure 4.73: Pressure Side % Chord vs. Impact Velocity, 1.33 mm FOD

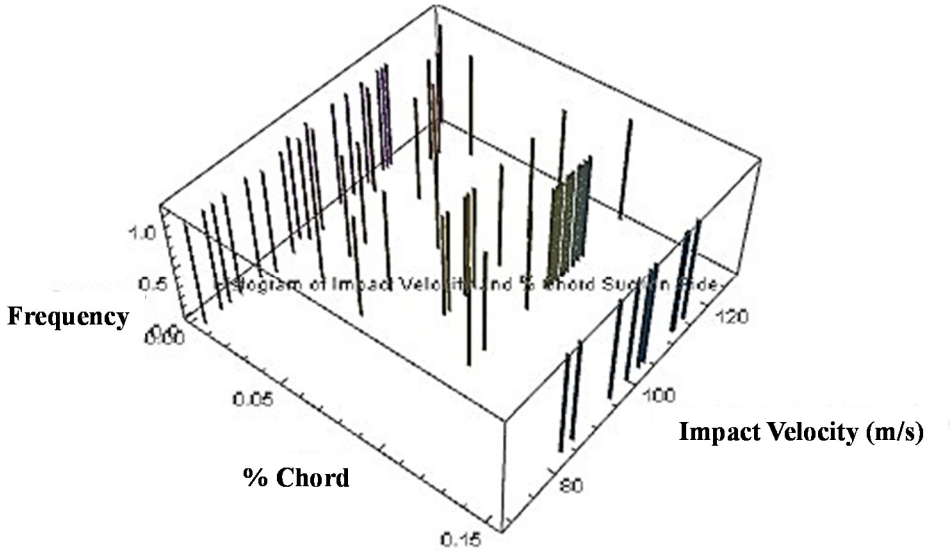


Figure 4.74: Suction Side % Chord vs. Impact Velocity, 3.2 mm FOD

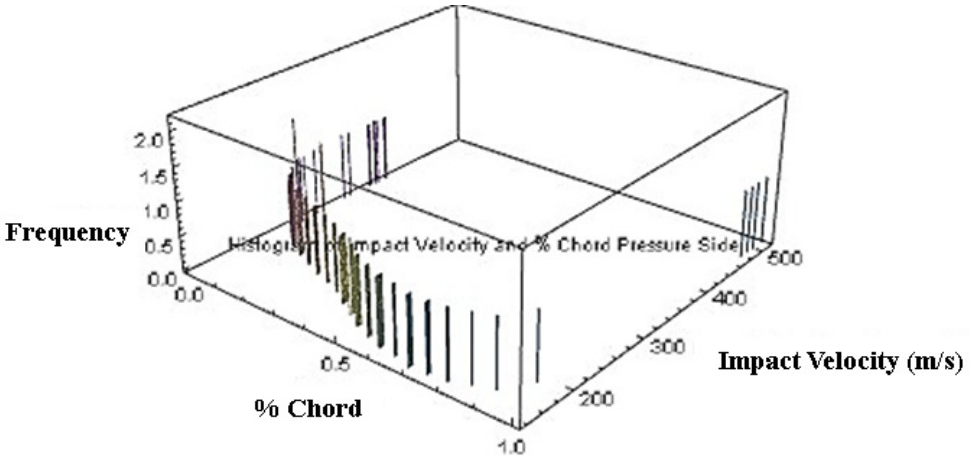


Figure 4.75: Pressure Side % Chord vs. Impact Velocity, 3.2 mm FOD

4.9.2.5 Damage Tolerance Interval Engine Inspections

The current paradigm for structural design of compressor blades is termed Safe Life or safe cyclic life [137]. This criteria is one of the oldest still practiced and one that still governs turbine and compressor design and certification activity [137, 138]. The Safe Life design criteria assumes a corrected, average life expectancy for parts under average usage and loading - the average life expectancy is corrected with a scatter factor - this factor is the ratio of test demonstrated crack free life to the minimum life of the component [137]. This design life criteria drives fixed, scheduled inspections [137]- for the F100-PW100 series engines on F-15s the original design life was based on the accumulation of 1800 Total Accumulated Cycles (TACs) that for the assumed usage of the aircraft in this thesis, ferry flights at constant spool rpm (75% N2 Speed, the code for the simulation for Crack Growth and LCC Estimation in Appendix F presents the implementation of the reduction of reference axial stress by means of N2 spool speed reduction) at an average altitude of 30,000 ft. lasting 2 hours, places the first scheduled engine teardown/overhaul inspection at 600 flight hours (from an average of 3 TACs per Flight Hour, average was estimated from the TACs and Flight hours at time of crash from an incident report of a Hellenic Air Force F-16D)[139, 140]. The 600 flight hours represent 300 takeoffs during what will henceforth in this thesis be referred to as the lifetime of the engine - the lifetime to be considered is up to the first scheduled engine teardown/overhaul of 300 takeoffs. Legacy F-15s averaged 270 flight hours per year up to the 1990s [141]. This defined engine lifetime is contextualized in takeoffs for FOD susceptibility: recall from Section 4.4 that the engine faces the riskiest regime for aspiration of ground debris during aircraft operation at the start of a runway during the early takeoff phase - typically in the first 3-4 seconds after the

engine is throttled to 100% power and the breaks are released. The decision was made to consider on-runway activities such as sweeping (removing by means of towed sweepers), runway debris inspections and unscheduled engine maintenance when significant structural damage has been signaled by sensors or by pilots up to, but not including, the scheduled teardown/overhaul inspection at the end of 300 takeoffs – note that by means of vibrations monitoring events such as the loss of a piece of blade are detected by the signaled loss of tuning at a rotor disk, pilots report even slight changes they sense and it is documented that engine fuel efficiency losses 0.5% – 1% are a sign of severe FOD strikes [17, 30, 150, 151]. The intent is to implement an interval engine inspection schedule protocol derived from Damage Tolerant design criteria - a paradigm shift from the current criteria of Safe Life that assumes that cracks will not grow beyond the First Engineering Crack, this size is reported in Figure 4-52 along with field cracks from FOD particle impacts that to near entirety are much larger [17, 128] - the current design criteria for blades is un-conservative as FOD damage to blades is pervasive and is not considered as design driver (Section 2.5).

Damage Tolerance design criteria works on the principle that "as-built" flaws or cracks exist on structural components and their design must tolerate/sustain their growth until repairs can be made. These "as-built" cracks are specifically termed in applied Fracture Mechanics as Equivalent Initial Flaw Size (EIFS) - they also represent the regressed, time zero initial sizes from cracks found during engine inspections (field cracks) - the point is precisely made in Figure 4-52, the FAA's First Engineering Crack is very likely and non-conservatively smaller than the EIFS regressed from the field cracks presented in that figure. In the context of Damage Tolerance design criteria an interval engine inspection schedule can be developed to find cracks whose sizes may drive a repair

or a replacement of the blade - this interval schedule protocol would provide a damage tolerant capability to the blade without any geometric redesign. Henceforth in this thesis the interval engine inspections will be also be referred to as damage tolerant interval engine inspections or DTA interval engine inspections. The interval inspection schedule to be considered in this thesis is presented in Figure 4-76.

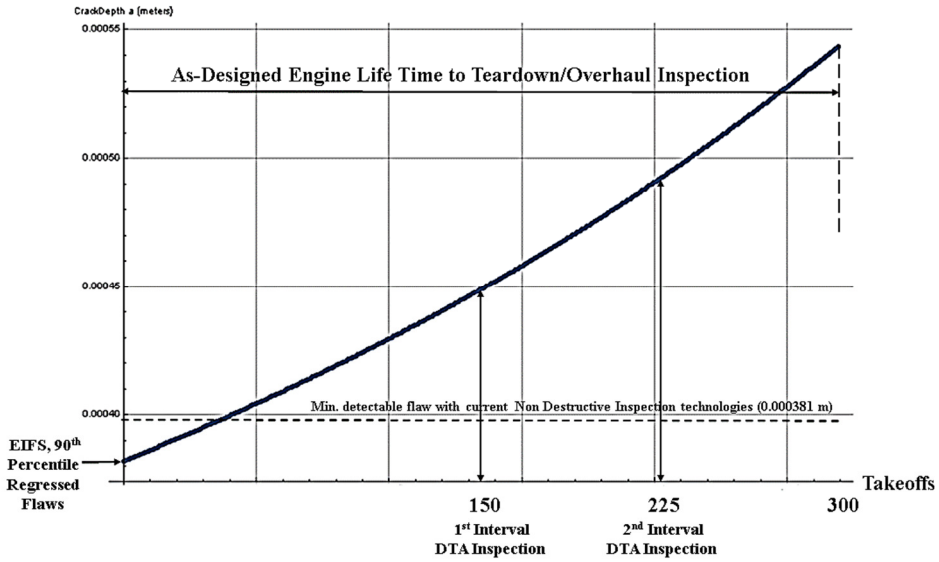


Figure 4.76: DTA Interval Engine Inspection Schedule

The DTA interval engine inspection schedule needs to be contextualized with engine stress cycles as Crack Growth Rate da/dN (Section 4.9.2.3) in the Fracture Mechanics discipline is typically not presented in crack length per takeoff. To convert takeoffs to stress cycles the frequency of the uniaxial load spectra must be known or approximated. Ritchie et al tested coupons of Ti-6Al-4V under simulated FOD impact conditions at varying stress frequencies and discovered that frequency has a negligible effect on da/dN [131] – those tests had to be done within a time frame much shorter than engine life, thus the stress frequencies are increased significantly. The author of this thesis chose a frequency of 25 Hz for the stress cycles that the modeled engine will be subjected to – this choice is within the range that Ritchie et al considered of 20 – 1500 Hz [131]. The

stress cycle increment that will be employed to load the 1st HPC rotor blades for each flight lasting two hours (Appendix F presents the implementation of the stress frequency as an incremental cycle and as the engine lifetime cycle totals for 300 takeoffs) is as follows:

$$N_{increment} = 2\ hours \cdot 60\,\frac{minutes}{hour} \cdot 60\,\frac{seconds}{minute} \cdot 25\ \frac{cycles}{second} = 180{,}000\ cycles \qquad (227)$$

4.9.2.6 Equivalent Initial Flaw Size & Critical Flaw Size

With the formulation for Stress Intensity (K) and Crack Growth Rate (da/dN) the Equivalent Initial Flaw Size (EIFS) and the critical flaw/crack size can regressed and estimated respectively.

When a FOD particle impacts a blade at a particular takeoff and radial location, be it on the mid-section, leading or trailing edge (see Figure 4-57), the penetration depth may exceed the depth from the naturally growing crack that emanated from the originally assumed "as-built" flaws (recall the LEFM validity threshold value of 0.00025 m calculated in Section 4.9.1) (see Figure 4-77). The assumed "as-built" flaw size may be too small and un-conservatively leads to crack growth that is artificially retarded because the smaller starting point provides more time to grow a crack – as there is finite engine lifetime, cracks could artificially not trigger repair or replace activity that is scheduled at specific takeoff intervals because the flaw size at those intervals (time, stress cycles or takeoffs) would be artificially smaller. A solution is to regress to time zero the "bumped-up" cracks at impact by means of a simple routine that loops in stress cycle increments back to a_0 (regressed flaw size). As impacts are random (in occurrence and location on a blade) during an engine lifetime many simulations of lifetime need to be run to create a vector of regressed flaw size – the 90th percentile of the regressed flaw size vector estimates the EIFS – Figure 4-78 presents the process by which EIFS is calculated – 500 lifetime simulations

were run for EIFS for the 1.33 and 3.2 mm diameter particles considered in this thesis, resulting in magnitudes of 0.000251067 and 0.000382119 m respectively.

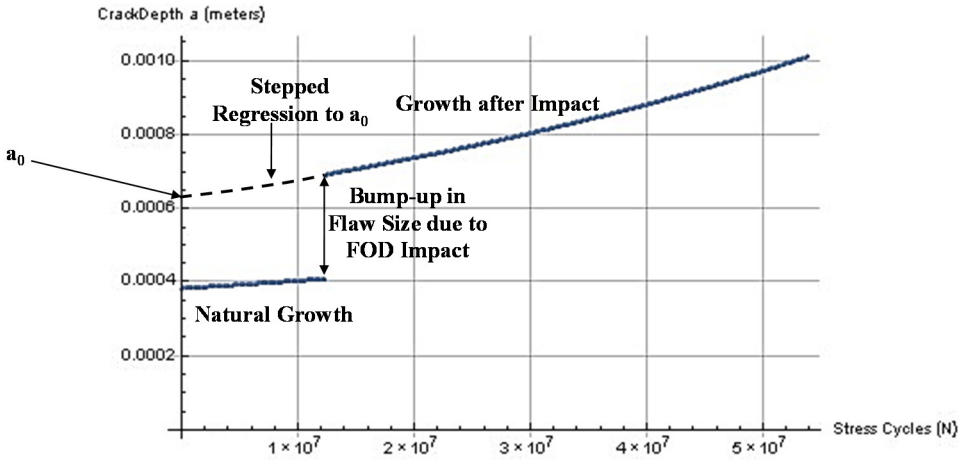


Figure 4.77: Flaw Size at Impact and Regression to Time Zero

Do[

$a_0 = a_{impact} = Particle\ Penetration\ Depth\ at\ Impact$

Do[

$da = \left(5.2 \cdot 10^{-12} \cdot \Delta K^{2.5} \cdot {K_{max}}^{0.67}\right) \cdot N_{increment}$;

$a_0 = a_0 - da$,

$\{i, 1, i_{impact}+1\}]$, $\{n, 1, n_{LifeTimes}\}]$

EIFS = Percentile[$\{a_0\}$,0.90];

Figure 4.78: Process to Estimate EIFS

The point at which fracture can occur is when the Stress Intensity reaches the Fracture Toughness (K_c) of the material. This limiting condition allows the calculation of the critical flaw size at the point of impact from Equations 194, 207 and 214 by solving for a_c (aac in simulation for Crack Growth and LCC Estimation code, Appendix F) as follows:

$$\frac{K_c}{\sigma_{max}} = \frac{K_c(t)\cdot(1+R)}{2\cdot\sigma} = \beta(a_c) \tag{228}$$

βs for airfoil mid-section and leading/trailing edge are described by Equations 195 and 208.

4.9.2.7 Repair/Replace Criterion

With the formulation for LEFM "large" crack growth the basic questions on the structural life of the LRU can be posed:

- When/how often should a blade be repaired?
- When/how often should a blade be replaced?

The answer to these questions can be had by implementing the following criterion (in the form of If statements in Figures 4-79 – 4-81) in the code for the simulation of Crack Growth and LCC Estimation (Appendix F) as a trigger for repair or replace activity – the simulation records when in the life of the blade the repairs or replacements occur.

```
LevelIIIDamageRandomSeed = RandomReal[];
(* Random Probability that a Level III CAAM Hazard,
from FAA Continued Airworthiness Assessment Methodologies (CAAM) process, |
may occur (has to be less than or equal to 4*10^-5) *)
If[EngineIntervalInspections == 1, (* Activates Interval Inspections *)
   (* Routine checks that if at 1st Interval Inspection the crack size is ≥
   than minimum detactable size (aaD) and <
   the local critical flaw size (aac) a Repair is required *)
   If[Ncycles == NcyclesIntervalInspection && aaD ≤ aa < aac,
      Nrepairs = Nrepairs + 1;
      aa = EIFSa,
      (* Checks that if within the aaD≤aa<aac range if a
      Level III CAAM Hazard has not occured the Blade will be Replaced,
      if the hazzard has occurred the Engine will be Replaced *)
      If[LevelIIIDamageRandomSeed ≥ 4 * 10^-5,
         If[Ncycles == NcyclesIntervalInspection && aa ≥ aac,
            Nreplacements = Nreplacements + 1;  aa = EIFSa],
         If[LevelIIIDamageRandomSeed < 4 * 10^-5 & Ncycles == NcyclesIntervalInspection &&
            aa ≥ aac,
            NEnginereplacements = NEnginereplacements + 1;
            aa = EIFSa]]];

   (* Routine checks that if Level III CAAM Hazard has not occurred and the 1st
   Interval Inspection has not occurred and the crack exceeds the critical
   flaw size (aac) a Blade Replacement is required *)
   If[LevelIIIDamageRandomSeed > 4 * 10^-5,
      If[Ncycles < NcyclesIntervalInspection && aa ≥ aac,
         Nreplacements = Nreplacements + 1;
         aa = EIFSa],
         (* Routine checks that if Level III CAAM Hazard has occurred and
         the 1st Interval Inspection has not occurred and the crack exceeds
         the critical flaw size (aac) an Engine Replacement is required *)
         If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5,
            If[Ncycles < NcyclesIntervalInspection && aa ≥ aac,
            NEnginereplacements = NEnginereplacements + 1;
            aa = EIFSa]]];
```

Figure 4.79: Blade Repair/Replace & Engine Replace Criterion

```
(* Routine checks that if Level III CAAM Hazard has not occurred and the 2nd
Interval Inspection has not occurred (1st Interval Inspection has occurred)
and the crack exceeds the critical flaw size (aac) a Blade Replacement
is required *)
If[LevelIIIDamageRandomSeed > 4 * 10 ^ -5,
   If[NcyclesIntervalInspection * 1.5 > Ncycles > NcyclesIntervalInspection && aa ≥ aac,
      Nreplacements = Nreplacements + 1;
      aa = EIFSa],
      (* Routine checks that if Level III CAAM Hazard has occurred and the 2nd
      Interval Inspection has occurred (1st Interval Inspection has occurred)
      and the crack exceeds the critical flaw size (aac) an Engine Replacement
      is required *)
      If[LevelIIIDamageRandomSeed ≤ 4 * 10 ^ -5,
         If[NcyclesIntervalInspection * 1.5 > Ncycles > NcyclesIntervalInspection &&
  aa ≥ aac,
            NEnginereplacements = NEnginereplacements + 1;
            aa = EIFSa]]];

(* Routine checks that if at 2nd Interval Inspection the crack is ≥
than minimum detactable size (aaD) and <
local critical flaw size (aac) a Repair is required *)
If[Ncycles == NcyclesIntervalInspection * 1.5 && aaD ≤ aa < aac,
   Nrepairs = Nrepairs + 1;
   aa = EIFSa,
   (* Checks that if at 2nd Interval Inspection if a Level III CAAM Hazard
   has not occured and the crack is ≥
   critical flaw size (aac) the Blade will be Replaced *)
   If[LevelIIIDamageRandomSeed ≥ 4 * 10 ^ -5,
      If[Ncycles == NcyclesIntervalInspection * 1.5 && aa ≥ aac,
         Nreplacements = Nreplacements + 1;
         aa = EIFSa],
         (* Routine checks that if Level III CAAM Hazard has occurred at 2nd
         Interval Inspection has and the crack exceeds the critical
         flaw size (aac) a Engine Replacement is required *)
```

Figure 4.80: Blade Repair/Replace & Engine Replace Criterion (cont.)

```
If[LevelIIIDamageRandomSeed < 4 * 10 ^ -5 && Ncycles == NcyclesIntervalInspection * 1.5 &&
  aa ≥ aac,
       NEnginereplacements = NEnginereplacements + 1;
       aa = EIFSa]]];

(* Routine checks that if Level III CAAM Hazard has not occurred and the 2nd
Interval Inspection has occurred and the crack exceeds the critical
flaw size (aac) a Blade Replacement is required *)
If[LevelIIIDamageRandomSeed > 4 * 10 ^ -5,
   If[Ntakeoffs * 2 * 60 * 60 * StressFrequency > Ncycles > NcyclesIntervalInspection * 1.5 &&
 aa ≥ aac,
      Nreplacements = Nreplacements + 1;
      aa = EIFSa],
      (* Routine checks that if Level III CAAM Hazard has occurred and the 2nd
      Interval Inspection has occurred and the crack exceeds the critical
      flaw size (aac) an Engine Replacement is required *)
      If[LevelIIIDamageRandomSeed ≤ 4 * 10 ^ -5,
         If[Ncycles > NcyclesIntervalInspection * 1.5 && aa ≥ aac,
            NEnginereplacements = NEnginereplacements + 1;
            aa = EIFSa]]],

 (* Routine checks that when Interval Inspections are not active up to but
 not including the engine lifetime,
 if a Level III CAAM Hazard has not occurred and the crack ≥
 critical flaw size (aac) the Blade will be Replaced*)
 If[EngineIntervalInspections == 0,
    If[LevelIIIDamageRandomSeed > 4 * 10 ^ -5,
       If[Ncycles < Ntakeoffs * 2 * 60 * 60 * StressFrequency && aa ≥ aac,
          Nreplacements = Nreplacements + 1;
          aa = EIFSa],
         (* Routine checks that when Interval Inspections are not active up to but
         not including the engine lifetime,
         if a Level III CAAM Hazard has occurred and the crack ≥
         critical flaw size (aac) the Engine will be Replaced*)
         If[LevelIIIDamageRandomSeed ≤ 4 * 10 ^ -5,
            If[Ncycles < Ntakeoffs * 2 * 60 * 60 * StressFrequency && aa ≥ aac,
               NEnginereplacements = NEnginereplacements + 1;
               aa = EIFSa]]]]]
```

Figure 4.81: Blade Repair/Replace & Engine Replace Criterion (comp.)

4.10 LIFE CYCLE COST MODEL

A buildup of models and simulation environments has been presented beginning with the characterization/quantification of damage to military jet engines and followed in sequence by models/simulations for on runway FOD, geometry external to engine, operational ground conditions, vortex formation/strength/location, debris aspiration from

ground to fan face, in-engine flow aero/thermos dynamics (Meanline Analysis), in-engine particle kinetics and crack growth. The question now becomes: by synergy of the physical and statistical models with suitable economic models can the overall cost trends for the blades be identified from the life cycle of the engine? Note that the simulations for particle impact and subsequent crack growth focus on a single engine, while the LCCs are scaled to fleet level by means of number aircraft (1,200 in fleet), number of engines per aircraft (2) and number of rotors on the HPC (13). A key assumption to estimate LCCs for any number of aircraft including a fleet is that the probability of impact once a particle passes the 1st HPC rotor is the same – the enablers to this assumption are that in military jet engines the HPC compressor rotor blade geometries change gradually and very little across those stages, it is assumed that the blade chord to pitch ratio (solidity) stays the same, additionally the particle axial velocity should not change significantly as its impulsive force comes from the meridional velocity that changes gradually and very little and lastly the HPC spool speed is the same – thus to account for the number of rotors and engines being modeled 2·13 suffices where appropriate (refer to the Crack Growth and LCC simulation code in Appendix F for details on how this assumption is employed to estimate LCCs and refer to Section 4.10 for the formulae for cost estimation).

Purposely missing to this point is how FOD on the runway is being rid of by means of detection/removal – that piece of the puzzle is critical because when it is seen through the prism of the Primary System (the blade) it may reveal the impact from the current and novel processes and technologies of detecting and ridding runway debris on the LCCs of the blades. The current paradigm for runway debris detection and removal is by visual inspections – two airport personnel inspect a runway twice per day (morning and evening)

by car. Additionally the current practice calls for one sweep of the runway with debris removing equipment. A new (albeit hybrid because it will retain the debris sweeping paradigm) paradigm will be considered where visual inspection will be replaced by automated inspections with sensing equipment. Typically manned/manual operations such runway or engine inspections are labor intensive and tend to dominate long term costs.

4.10.1 INSPECTIONS/REMOVALS & LCCS, VISUAL INSPECTIONS

4.10.1.1 Inspection/Removal Model, Visual Inspections

Currently the inspection of a runway is done visually on a vehicle by two personnel driving at 20 mph [30]. This inspection process is able to find and remove only 3% - 4% of FOD on the runway per operational 24 hour day [30]. Runways at military bases in the United States are inspected 1 time per operational 24 hour day in the same manner as their commercial counterparts, visually on a moving vehicle with similar inspection/removal performance – Moffett Field in California goes through a single, visual inspection every morning [70]. In the context of this thesis the assumption will be made that the current paradigm for FOD management at an airport requires 2 daily visual inspections performed by two personnel on a moving vehicle with the ability to detect and remove 3% - 4% of runway debris per operational 24 hour day. The literature also supports the assumption that commercial and military runways are swept with mobile/towed debris removing equipment 1 time per operational 24 hour day – mobile/towed debris removing equipment is mandated by the FAA to remove a minimum of 90% of FOD in the path of the sweep at a minimum speed of 15 mph [30, 41].

Following the assumption in this thesis that airport FOD management is a hybrid of commercial and military practices it will be assumed that the model for detection and

removal of debris on the runway depends on a hybrid of the military FOD density presented in Section 4.2.2 and FOD runway distribution presented in Section 4.2.3. Recall from Equations 9a and 9b and Figure 4-3 that the total number of debris particles on the runway, their density (number of particles per area) and distribution are related. As debris is detected and immediately removed by airport personnel the total number of particles on the runway is reduced in an operational 24 hour day. Likewise as sweeping equipment is deployed on the runway it removes an amount of FOD for the operational 24 hour day. Recall from Section 2.8.2.1 that a model was proposed to gage the efficacy of eradicating/ridding runway FOD, the Measure of Eradication (M_oER) formula (Equation 1) will be employed to reduce the quantity of FOD on the runway after visual inspections and sweeping takes place. The assumption is made that the reduced quantity of FOD is distributed uniformly and following the model proposed in Section 4.2.3 - this assumption is supported by McCreary's work that presents an overlay of FOD reported at the end of an operational day on a runway that is uniformly distributed [30]. The proposed model is as follows:

$$2 \cdot M \cdot N + M + N + 1 = \rho_{runway} \cdot L_{runway} \cdot W_{runway} \cdot \left\{ \frac{24 - N_{sweep} \cdot \frac{L_{runway} \cdot W_{runway}}{AreaSweep/hr} \cdot \eta_{sweep} - N_{inspections} \cdot T_{visualinspection} \cdot \eta_{visual} \cdot \eta_{removal}}{24} \right\} \quad (229)$$

Where,

Nsweep - number of times the debris removal process and/or technology is engaged, in this case sweeping by mobile/towed debris removing equipment (FOD*BOSS)

T_{sweep} – duration of debris removal process and/or technology procedure, in this case a Triplex System (uses three FOD*BOSSes towed by one vehicle) is assumed to be the

choice for a large military runway as it can sweep 300,000 m^2/hour, thus the time to sweep the runway once is $L_{runway} \cdot W_{runway}/(AreaSweep/hr)$

η_{sweep} – efficacy of removal process and/or technology, in this case the baseline efficacy of removal of debris by sweeping is 0.90 mandated by the FAA [41]

$N_{inspections}$ - how many times an inspection process and/or technology is performed, in this case number of visual inspections – baseline two per operational 24 hour day is assumed

$T_{visualinspection}$ – duration of inspection process and/or technology procedure, in this case duration of one runway visual inspection – the baseline duration is 0.5 hours [30]

η_{visual} – detection efficacy of inspection process and/or technology - the baseline efficacy of inspection detection is 0.8 [30]

$\eta_{removal}$ – efficacy of removal during inspection process and/or technology, in this case the assumption is made that inspections personnel remove all that is detected, $\eta_{removal} = 1$

AreaSweep/hr – the sweeping equipment area sweep per hour, in this case a Triplex System (uses three FOD*BOSSes towed by one vehicle) can sweep 300,000 m^2/hour

d – runway width-wise FOD particle pitch

ρ_{runway} – on runway FOD density

Equation 6 relates the runway width counter N for number of FOD particles to pitch by

$$N \cdot d = W_{runway}$$

Equation 7 relates the runway length counter M for number of FOD particles to pitch by

$$M \cdot d\sqrt{3} = L_{runway}$$

Reducing number of variables by means of Equations 6 and 7 and particle pitch d

$$M = {N \cdot L_{runway}} \Big/ {W_{runway} \cdot \sqrt{3}}$$

Solving Equation 229 for N,

$$N = \left(\frac{1}{L_{runway}} \left(-\frac{L_{runway}}{4} + 0.43301 \cdot W_{runway} \left(-1 + \left(\left(1 + \right. \right. \right. \right.$$

$$\left. \left. \left. \frac{L_{runway}}{{W_{runway}}^2} \left(-3.4641 \cdot W_{runway} + L_{runway} \left(\frac{1}{3} + \right. \right. \right. \right.$$

$$\left. \left. \left. {W_{runway}}^2 0.19245 \cdot \left(24 - \right. \right. \right. \right.$$

$$\left. \left. \left. \left(N_{sweep} \frac{L_{runway} \cdot W_{runway}}{\frac{\text{AreaSweep}}{\text{hr}}} \eta_{sweep} + \right. \right. \right.$$

$$\left. \left. \left. \left. N_{inspection} T_{visualinspection} \eta_{visual} \right) \right) \rho_{runway} \right) \right) \right) \Bigg)^{0.5} \Bigg) \Bigg) \Bigg) \Bigg) \Bigg) \tag{230}$$

With a parametric formulation for the FOD runway width counter N a modified Equation 27 will estimate the probability that a particle will be engulfed by the ground vortex core of radius r_c with the inclusion of inspections and runway sweeps – this is the probability the particle will be inside the ground vortex where aspiration conditions are met according to Section 4.5.2.

$$P_{EngulfedFOD_{visualinspections}} = \frac{N \cdot \left\{ D_{VortexCore} - \frac{D_{FOD}}{2} \right\}}{W_{runway}} \tag{27a}$$

Where,

$D_{VortexCore}$ – is twice the vortex core radius r_c

To develop formulation that approximates the scaling for r_c from Murphy's work on ground vortex ingestion to full size considered in this thesis assumptions are made [30]:

- A key scaling parameter in Murphy's work is non-dimensional vortex strength

$$\Gamma^* = \frac{\Gamma}{U_i \cdot D_l} \tag{231}$$

Where,

Γ- vortex flow circulation

U_i – inlet velocity

D_l – inlet diameter

- In Murphy's work Γ^* is a dependent variable thus by dynamic similarity the assumption is made that the full scale non-dimensional vortex strength of the vortices on the runway of this thesis have the same magnitude

$$\Gamma^*_{scale} = \frac{\Gamma_{scale}}{U_{i_{scale}} \cdot D_{l_{scale}}} = \frac{\Gamma_{fullscale}}{U_{i_{fullscale}} \cdot D_{l_{fullscale}}} = \Gamma^*_{fullscale} \qquad (232)$$

- An assumption that was made implicitly in Section 4.5.3 is that ground vortices are plane potential flows – they behave like solid bodies inside the vortex core (often called forced vortices) - their tangential velocity ($V_r = 0$, there is no radial flow) is described as

$$V_\theta = k \cdot r \qquad (233)$$

If a velocity potential is assumed as follows,

$$\varphi = k \cdot r^2 \cdot \theta \qquad (234)$$

Then from the polar coordinates formulation,

$$V_\theta = \frac{1}{r} \cdot \frac{\partial \varphi}{\partial \theta} = k \cdot r \qquad (233a)$$

Velocity is in m/s k is made to be equal to the angular velocity of the flow Ω

- From Equation 234 the vortex circulation can be calculated by the path integral of the potential total derivative in polar coordinates

$$\Gamma = \oint_C d\varphi = \int_0^{2\pi} \frac{\partial \varphi}{\partial \theta} \cdot d\theta + \int_0^r \frac{\partial \varphi}{\partial r} \cdot dr = \int_0^{2\pi} k \cdot r^2 \cdot d\theta = 2\pi \cdot \Omega \cdot r^2 \qquad (235)$$

- Recalling that a premise in this thesis was that the inlet velocities where of the same order as Murphy's because inlet Mach numbers were matched (Section 4.7.3), assuming that the angular velocity of scale and full scale flows is the same and recalling that Murphy's model was 1/30th scale

$$\frac{\Gamma_{scale}}{U_i \cdot D_{l_{scale}}} = \frac{\cancel{2\pi \cdot \Omega} \cdot {r_{c,scale}}^2}{\cancel{U_t} \cdot D_{l_{scale}}} = \frac{\cancel{2\pi \cdot \Omega} \cdot {r_{c,fullscale}}^2}{\cancel{U_t} \cdot D_{l_{fullscale}}} = \frac{\Gamma_{fullscale}}{U_i \cdot D_{l_{fullscale}}} \tag{232a}$$

The scaling of the vortex core is revealed - Equation 27a can now parametrically calculate the probability that the particle will be inside the full size ground vortex core where aspiration conditions are met

$$r_{c,fullscale} = r_{c,scale} \cdot \sqrt{30} \tag{236}$$

4.10.1.2 LCC Model, Visual Inspections

Recall from Section 4.9.2.5 that legacy F-15s flew an average of 270 hours per year that with an average ferry flight considered in this thesis of 2 hours an average of 135 takeoffs per year is calculated – the result is that on average the aircraft being resembled has a takeoff every 2.704 days. This number is a key parameter to calculate costs in the life cycle of the blade as it relates takeoffs to the daily sweeps and runway inspections (refer to the code for the simulation of Crack Growth and LCC Estimation, Appendix F). The sweeping equipment acquisition cost at the base where all the aircraft of the engine type modeled in this thesis (F-15s and F-16s) take off is shared – the assumption is made that there are 1200 such aircraft with two engines a piece (the assumption is supported by the current number of F-16s and F-15s in the U.S. Air Force [143]). Inspection and sweep vehicle usage cost per mile and labor costs for two personnel from the public domain are accounted for as well – the accounting takes into account how long a single sweep of the runway takes and how many sweeps are done along with the labor and vehicle usage costs

incurred for the time duration of these daily activities/processes [144, 145]. As visual inspections take place on a moving vehicle a speed of 30 mph is chosen based on McCreary's reported numbers for inspection vehicle speed [30]. The cost model is as follows:

$$\mathrm{LCC}_{visualinspection} = \frac{RSAC}{1200} + \sum_{nn=1}^{N_{takeoffs}} 2.704 \left\{ N_{sweep} \cdot L_{runway} \cdot \frac{W_{runway}}{W_{sweeper}} \cdot CVUM + \right.$$

$$\frac{L_{runway} \cdot W_{runway}}{\frac{AreaSweep}{hr}} \cdot N_{sweep} \cdot 2 \cdot BDL + N_{inspection} \cdot T_{visualinspection} \cdot IVS \cdot CVUM +$$

$$\left. N_{inspection} T_{visualinspection} \cdot 2 \cdot BDL \right\} \quad (237)$$

Where,

RSAC (Runway Sweep Acquisition Cost) – cost to acquire sweeping equipment, for this thesis a Triplex System that uses three FOD*BOSSes towed by one vehicle will be assumed with each of the three units costing $7,000

CVUM (Cost of Vehicle Use per Mile) – cost of vehicle usage per mile, the literature lists that for a small vehicle, likely to be used for runway inspections, the cost per mile is $0.1375 [144]

IVS – inspection vehicle speed (in mph)

BDL (Burdened Direct Labor) – is the burdened direct labor cost per work hour from the DOD of $111,426.95/2080 in 2015 US dollars [145]

4.10.2 Inspections/Removals & LCCs, Auto. Inspections

4.10.2.1 Inspection/Removal Model, Auto. Inspections

As was mentioned before a new (albeit hybrid because it will retain the debris sweeping paradigm) paradigm will be considered where visual inspection will be replaced

by automated inspections with sensing equipment on the runway. McCreary's work presents a case for automated scanning because it has a higher sensitivity (90%– 99% probability of detection vs. 80% for visual inspections) and due to the short time between scans (can be dialed down to minutes or seconds, nearly negligible) the assumption can be made that automated FOD sensing systems are in continuous operation [30,148]. Table 4.7 lists the three most widely deployed automated FOD detection technologies whose capabilities and costs are in the public domain – all FOD detection systems are mandated by the FAA to detect debris much larger (>2 cm) than considered in this thesis [75, 146, 147]. As can be appreciated from the table two technologies are able to detect the size of FOD considered in this thesis (1.33 and 3.2 mm diameter spherical steel particles). Because one of the systems works on much larger debris a compromise/assumption is made to McCreary's range of detection capability – the range of probability of detection will be 0.65 to 0.74, a knockdown in capability of 0.25. Per Table 4.7 another assumption can be made: the cost of the more sensitive systems is higher, thus a linear relationship between sensitivity and cost will be assumed (to review the detection sensitivity-cost relationship refer to the code for the simulation of Crack Growth and LCC Estimation, Appendix F). The assumption is made that the system to be considered/modeled is a fixed, runway lights collocated technology whose detection sensitivity could be any value in the 0.65 to 0.74 range whose cost to acquire is indexed by the sensitivity to the dollar amounts in Table 4.7. Per the literature for fixed systems their largest cost contributor is acquisition – once installed they require very little power and maintenance is typically built into the acquisition contract [146, 147].

Table 4.7: Automated FOD Detection Systems Capability and Cost

System (vendor, if provided)	Mode of Sensing	Detection Probability of FAA mandated FOD (resolution)	Cost (2015 U.S. dollars)	Source
FODetect (XsightSystems)	Optical-radar sensors collocated on runways. Hybrid integrated optic sensor with NIR illumination and a millimeter-wave radar sensing.	≥ 90% (can detect rivets, <5 mm diam.)	$1.73 million	"Logan Airport installs runway scanning system", Boston Globe.
iFerret™ (The Stratech Group Limited)	Optical system, employs high mast mounted cameras.	≥ 94% (can detect objects 4 cm diam.)	$18.44 million	"Stratech's iFerret™ Wins in Hong Kong", Stratech Systems Limited.
Pavemetrics™ Laser Foreign Object Debris (LFOD) detection system (Pavemetrics)	Mobile, laser FOD detection.	≥ 90% (can detect objects 2 mm diam.)	$0.763 million	"Laser-Based Fully Automated FOD Detection", Pavemetrics™.

Modeling inspection/removal of FOD on the runway for automated inspection is very similar to that of visual inspections. What changes is how the number of scans (term scan avoids confusion with runway sweeping) and the length of a scan are calculated. Automated FOD detection systems perform a number of scans per unit of time, a scan rate – it will be assumed that because takeoff activity is lower than in a commercial airport the number of scans in an operational 24 hour day will be fewer but lengthier in time followed by a fixed amount of time to remove debris ($T_{removal}$) that was detected and GPS located by the system – all the systems that were researched for Table 4.7 provide the GPS location on the runway where FOD is detected. The number of scans in an operational 24 hour day and the length of scan is related to the scan rate by:

$$N_{scan} = 24 \cdot ScanRate \tag{238}$$

$$T_{scan} = \frac{1}{ScanRate} - T_{removal} \tag{239}$$

The formulation for automated inspection and removal of debris from the runway in an operational 24 hour day is under the same assumption as the current paradigm - the reduced quantity of FOD is distributed uniformly and following the model proposed in Section 4.2.3. The formulation is as follows:

$$2 \cdot M \cdot N + M + N + 1 = \rho_{runway} \cdot L_{runway} \cdot W_{runway} \cdot$$

$$\left\{\frac{24 - N_{sweep} \cdot \frac{L_{runway} \cdot W_{runway}}{\frac{AreaSweep}{hr}} \cdot \eta_{sweep} - N_{scan} \cdot T_{scan} \cdot \eta_{scan} \cdot \eta_{removal}}{24}\right\} = \rho_{runway} L_{runway} W_{runway} \cdot$$

$$\left(\frac{24 - N_{sweep} \frac{L_{runway} \cdot W_{runway}}{\frac{AreaSweep}{hr}} \eta_{sweep} - 24 ScanRate\left(\frac{1}{ScanRate} - T_{removal}\right) \eta_{scan} \eta_{removal}}{24}\right) \quad (240)$$

Solving Equation 240 for N,

$$N = \left(\frac{1}{L_{runway}} \cdot \left(-\frac{L_{runway}}{4} + 0.43301 \cdot W_{runway} \cdot \left(-1 + \left(\left(1 + \right.\right.\right.\right.\right.$$

$$\left.\left.\left.\left.\left.\frac{L_{runway}}{{W_{runway}}^2}\left(-3.4641 \cdot W_{runway} + L_{runway} \cdot \left(\frac{1}{3} + \right.\right.\right.\right.\right.\right.$$

$$\left.\left.\left.\left.\left.{W_{runway}}^2 \cdot 0.19245 \cdot \left(24 - \right.\right.\right.\right.\right.$$

$$\left.\left.\left.\left.\left.\left(N_{sweep} \frac{L_{runway} \cdot W_{runway}}{\frac{\text{AreaSweep}}{\text{hr}}} \eta_{sweep} + 24 \cdot ScanRate \cdot \left(\frac{1}{ScanRate} - \right.\right.\right.\right.\right.$$

$$\left.\left.\left.\left.\left.\left.\left.T_{removal}\right) \eta_{scan} \cdot \eta_{removal}\right)\right) \cdot \rho_{runway}\right)\right)\right)\right)^{0.5}\right)\right)\right)\right) \quad (241)$$

Where,

ScanRate – number of runway scans per unit time, for this thesis three scan rates will be considered: 0.33/hour, 0.5/hour and 1/hour (1 scan every 3, 2 and 1 hour respectively)

$T_{removal}$ – duration of debris removal process and/or technology procedure, in this case the duration will be fixed to 0.5 hours

η_{scan} – detection efficacy of automated inspection process and/or technology - the efficacy of inspection detection has a uniform distribution with a range of 0.65 – 0.74

$\eta_{removal}$ – efficacy of removal during automated inspection process and/or technology, in this case the assumption is made that dispatched personnel are burdened with more detected debris and during the fixed removal time therefore $\eta_{removal} = 0.90$

As was mentioned before with a parametric formulation for the FOD runway width counter N a modified Equation 27 will estimate the probability that a particle will be engulfed by the ground vortex core of radius r_c (recall the vortex core radius for full scale is calculated following the procedure in Section 4.10.1.1)with the inclusion of automated inspections and runway sweeps – this is the probability the particle will be inside the ground vortex where aspiration conditions are met according to Section 4.5.2.

$$P_{EngulfedFOD_{automatedinspections}} = \frac{N \cdot \left\{ D_{VortexCore} - \frac{D_{FOD}}{2} \right\}}{W_{runway}} \quad (27b)$$

4.10.2.2 LCC Model, Auto. Inspections

The model for the LCCs from runway sweeping and automated inspections functions under the same assumptions as the cost model for visual inspections in Section 4.10.1.2. The only salient change is that because automated inspection technologies scan the runway 24 hours a day continuously personnel have to be on call when debris is detected – that is the burden of the technology that is often overlooked, for this thesis a two personnel will be on call during the operational 24 hour day (may be three 8 hour shifts). The acquisition cost of the detection technology must be accounted for as well – again that cost is shared amongst 1,200 aircraft that takeoff from the same runway.

$$\mathrm{LCC}_{automatedinspection} = \frac{RSAC}{1200} + \frac{CAAFDS}{1200} + \sum_{nn=1}^{N_{takeoffs}} 2.704 \left\{ N_{sweep} \cdot L_{runway} \cdot \frac{W_{runway}}{W_{sweeper}} \cdot CVUM + \frac{L_{runway} \cdot W_{runway}}{\frac{AreaSweep}{hr}} \cdot N_{sweep} \cdot 2 \cdot BDL + 24 \cdot 2 \cdot BDL \right\} \quad (242)$$

Where,

RSAC (Runway Sweep Acquisition Cost) – cost to acquire sweeping equipment, for this thesis a Triplex System that uses three FOD*BOSSes towed by one vehicle will be assumed with each of the three units costing $7,000

CVUM (Cost of Vehicle Use per Mile) – cost of vehicle usage per mile, the literature lists that for a small vehicle, likely to be used for runway inspections, the cost per mile is $0.1375 [144]

CAAFDS (Cost of Acquisition of Automated FOD Detection System) – cost of acquisition of detection technology, whose detection sensitivity could be any value in the 0.65 to 0.74 range and whose cost to acquire is indexed by the sensitivity to dollar amounts in Table 4.7

BDL (Burdened Direct Labor) – is the burdened direct labor cost per work hour from the DOD of $111,426.95/2080 in 2015 US dollars [145]

4.10.3 LSP LCC MODEL

The assumption is made that the military operator of the 1,200 aircraft with the engine modeled in this thesis will make the capital investment of equipment to perform Laser Shock Peening on the compressor blades – based on Shukla's LSP acquisition, treatment and process costs it appears that the investment is small and the payoff high if much more expensive engine components can be saved or life extended (see Table 4.8) [149]. The assumption is made that any acquisition cost or process time is equally likely from those presented in Table 4.8, but the clean-up and change-over times are constant. The assumption of keeping the LSP treatment work "in-house" has a consequence for modeling, the understood and documented DOD labor costs are applied to the process,

clean-up and change-over times for each of the 38 blades on the 1st HPC rotor disk [145]. Equation presents the model for LCC costs of LSP treatments.

Table 4.8: LSP Acquisition and Application Costs

Capital Costs (2015 US dollars)	Process Time Labor (minutes, per blade)	Clean-up Time Labor (minutes, per blade)	Change-over Time Labor (minutes, per blade)
$66,976 - $109,702	1-10	5	10

$$LCC_{LSP} = \frac{CALSPM}{1200} + n_{rotorblades} \cdot [(LSPTreatment \cdot LSPPT + LSPCUT + LSPCOT)/ 60 \cdot BDL] \quad (243)$$

Where,

CALSPM (Cost of Acquisition of LSP Machine) – cost to acquire LSP equipment

LSPPT – LSP process time labor (in minutes)

LSPCUT – LSP clean up time labor (in minutes)

LSPCOT – LSP change over time labor (in minutes)

BDL (Burdened Direct Labor) – is the burdened direct labor cost per work hour from the DOD of $111,426.95/2080 in 2015 US dollars [145]

4.10.4 Engine Interval Inspections, Repair/Replace LCCs

The LCCs of interval engine inspections, LRU (blade) repairs and replacements and engine replacements if an engine is severely damaged (as defined by the FAA an event such as a debris impact at a blade that crosses the threshold for per-flight, short-term risk of less than one Level 3 event in 25,000 flights (4x10-5) - a Level 3 event carries serious consequences including permanent loss of thrust or power to the propulsion system [152]) along with their associated material and labor costs will be calculated employing Meadows' methodology derived from the Boeing Dependability Cost Model [153].

Additionally, the cost (a cost delta) of fuel efficiency losses due to FOD strikes in the range of 0.5% – 1% will be included in the aggregate LCC costs [30].

The formulation for the damage tolerant interval engine inspections, LRU repairs/replacements and engine replacements comes from Meadows' work [153]. The formulation is presented first, it follows Meadows' naming convention with slight changes (this convention is adhered to in the code for the simulation of Crack Growth and LCC Estimation, Appendix F). The formulation for the costs associated with engine efficiency losses due to FOD impacts are approximated. The cost to replace an engine of $2.46 million (in US dollars) was calculated by the RAND cost methodology [155]. Definitions for terms follow after. The full implementation of the costing formulae resides in code for the simulation of Crack Growth and LCC Estimation in Appendix F. In Equations 244 – 250 the denominator is the calculation for the labor hours at the interval inspection, in this case 300 hours.

$$SCHINCOST = \frac{FLTS \cdot QTY \cdot AVEFLTHR \cdot SCHMNTINSMM \cdot BDL}{\left(\frac{1}{2} \cdot N_{takeoffs} \cdot AVEFLTHR\right)} \tag{244}$$

$$SCHCALrepairs = \frac{LifeTimerepairs \cdot (FLTS \cdot NA \cdot QTY \cdot AVEFLTHR \cdot RECMHRSrepairs \cdot BDL)}{\left(\frac{1}{2} \cdot N_{takeoffs} \cdot AVEFLTHR\right)} \tag{245}$$

$$SCHEDCAMATrepairs = \frac{LifeTimerepairs \cdot (FLTS \cdot NA \cdot QTY \cdot AVEFLTHR \cdot SCHCAMATrepairs)}{\left(\frac{1}{2} \cdot N_{takeoffs} \cdot AVEFLTHR\right)} \tag{246}$$

$$Lreplacements = LifeTimereplacements \cdot (FLTS \cdot NA \cdot QTY \cdot AVEFLTHR \cdot RECMHRSreplacements \cdot BDL) / \left(\frac{1}{2} \cdot N_{takeoffs} \cdot AVEFLTHR\right) \tag{247}$$

$$MATreplacements = LifeTimereplacements \cdot (FLTS \cdot NA \cdot QTY \cdot AVEFLTHR \cdot CAMATreplacements) / \left(\frac{1}{2} \cdot N_{takeoffs} \cdot AVEFLTHR\right) \tag{248}$$

$$LEnginereplacements = LifeTimeEnginereplacements \cdot (FLTS \cdot NA \cdot QTY \cdot AVEFLTHR \cdot MHRSEnginereplacements \cdot BDL)/\left(\frac{1}{2} \cdot N_{takeoffs} \cdot AVEFLTHR\right) \quad (249)$$

$$MATEnginereplacements = LifeTimeEnginereplacements \cdot (FLTS \cdot NA \cdot QTY \cdot AVEFLTHR \cdot CAMATEnginereplacements)/\left(\frac{1}{2} \cdot N_{takeoffs} \cdot AVEFLTHR\right) \quad (250)$$

$$CFF = MilitaryThrust \cdot SpecificFuelConsumption \cdot AVEFLTHR \cdot 0.3261 \cdot CurrentPriceofJetFuel \quad (251)$$

$$CIDEED = (0.005 + RandomRealNumber \cdot 0.005) \cdot 2 \cdot CFF \cdot (Ntakeoffs - Impacti)) \quad (252)$$

Where,

FLTS - Flights per year per aircraft

QTY - Component quantity per Aircraft,

SCHMNTINSMM - Scheduled inspection labor at the 1st and 2nd interval engine inspection

BDL (Burdened Direct Labor) – is the burdened direct labor cost per work hour from the DOD of $111,426.95/2080 in 2015 US dollars [145]

$N_{takeoffs}$ – Total takeoffs

AVEFLTHR – Average flight hours per flight

LifeTimerepairs – Number of LRU (blade) repairs in an engine lifetime

LifeTimereplacements – Number of LRU (blade) replacements in an engine lifetime

LifeTimeEnginereplacements – Number of engine replacements during engine lifetime

RECMHRSrepairs – Rectification man hours at the interval engine inspections for repairs

SCHCALrepairs - Scheduled Corrective Action labor costs for repairs

SCHINCOST – Scheduled inspection labor cost, the cost of the interval engine inspection

SCHCAMATrepairs - Scheduled Corrective Action material cost

SCHEDCAMATrepairs - Scheduled Corrective Action cost

RECMHRSreplacements - Rectification man hours for LRU replacements

Lreplacements – Labor costs of LRU replacements

CAMATreplacements – LRU replacement material cost

MATreplacements – LRU replacement cost

MHRSEnginereplacements - Man hours for an engine replacement

LEnginereplacements – Labor costs for engine replacement

CAMATEnginereplacements - Engine replacement material cost

MATEnginereplacements – Engine replacement cost

CFF – Cost of fuel per flight

CIDEED – Cost increase due to engine efficiency degradation

5 CASE STUDY

5.1 ONE EFFORT, TWO LESSONS

There are dominant lessons to be learned from the research effort of this thesis:

- Nature of particle impact at the rotor blades – location and velocity on the airfoil
- LCC and LRU replacement impact by FOD and its elimination

The following sections will present a case study for impact and ensuing structural/economic detriment at the 1st HPC rotor blades of two particle sizes, 1.33 and 3.2 mm diameter spheres of mild carbon steel – recall that the research objective of this thesis is to determine if FOD elimination, the minimization of the FOD-exacerbated HCF failure mode that leads to 1st HPC rotor blade replacements and/or catastrophic engine losses while reducing or marginally increasing LCCs, is achievable and if so by what mix of technologies, processes and/or design changes at the primary and support systems.

5.1.1 EXPERIMENTAL CONDITIONS

The experimental conditions that will drive the case study are as follows:

- Crosswind conditions are not considered and the trajectory plane of FOD particles is 2-D and has an incidence of 0° with respect to engine centerline axis (trajectory is on x-z plane where x is measured aft of engine centerline axis located at center engine inlet, z is measure vertically from this same location)
- Two FOD particle sizes, 1.33 and 3.2 mm diameter spheres of mild carbon steel with a density of 7850 kg/m^3
- The engine model is geometrically and performance-wise similar to the F100-PW100 engine family (Meanline Analysis Model presented in Section 4.7)

- Aircraft runway operation, early in the takeoff phase at a Standard Day (288.15 °K, 101.325 kPa and 1.225 kg/m^3 for ambient temperature, pressure and air density) at Sea Level conditions with inlet Mach numbers ranging 0.45 to 0.5.
- Fan and 1st HPC spool speeds N1 and N2 of 10,000 and 14,000 rpm respectively during takeoff at 100% engine power setting
- Numerical/analytical experiments will match Murphy's experimental condition for inlet operation under a Headwind (U_∞) and Rolling Ground (U_G) equal in magnitude [30] – Murphy chose the condition to rid the influence of boundary layers in his laboratory experiments – boundary layers are larger than aircraft on the runway the assumption is made that the effect of Headwind is dominant
- Turbomachinery sized at Mach 0.475 with Meanline model (see Figure 5-1)

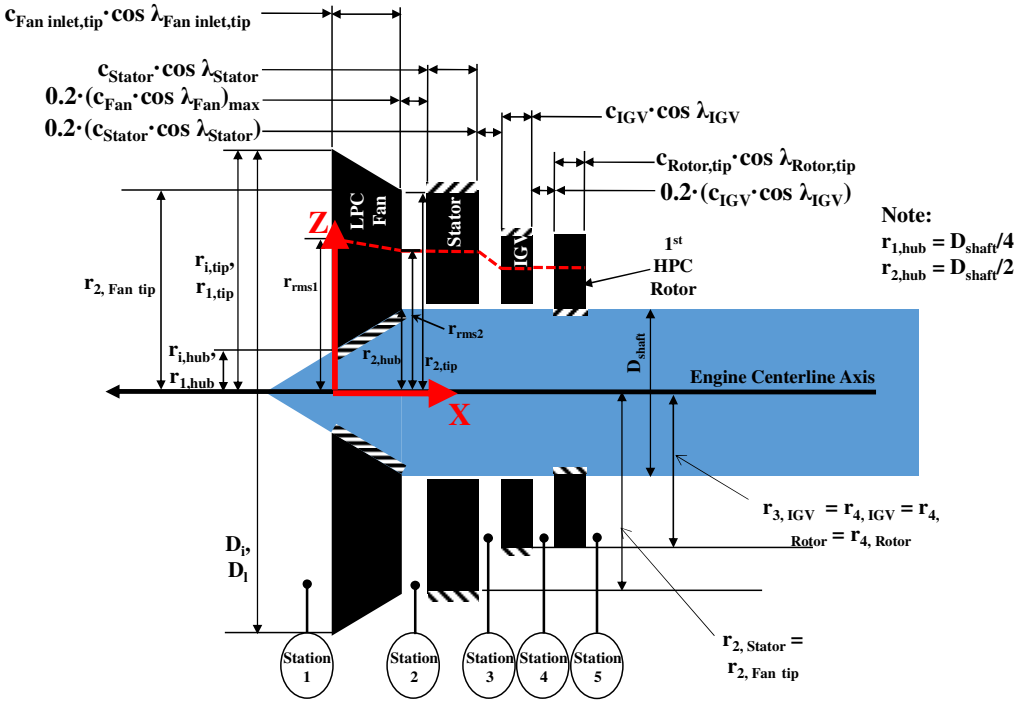


Figure 5.1: Fixed Engine Geometry

Where,

Spacing between stations is 0.2 * chord at tip * cosine (twist at tip)

D – Shaft diameter, 0.42 m

$r_{1,\,tip}$ - r at fan inlet (D_l/2), 0.44 m

$c_{Fan\ inlet,tip}$ – Fan chord at inlet tip, 0.0762667 m

$\lambda_{Fan\ inlet,\,tip}$ - Twist angle at tip of fan at inlet, 0.097117 radians measured counter-clockwise from stagger angle at r_{rms} (see Figure 4-30)

c_{Stator} – Stator chord, 0.070421 m (stators have fixed chord, measured clockwise from engine centerline axis)

λ_{Stator} - Twist angle at stator, 0.509946 radians (stators have fixed stagger)

c_{IGV} – IGV chord, 0.058090 m (IGVs have fixed chord, measured clockwise from engine centerline axis)

λ_{IGV} - Twist angle at IGV, 0.305109 radians (IGVs have fixed stagger)

$c_{Rotor,tip}$ – Rotor chord at tip, 0.023740 m

$\lambda_{Rotor,tip}$ - Twist angle at tip of rotor, 0.085565 radians measured counter-clockwise from stagger angle at r_{rms} (see Figure 4-30)

$r_{2,\,Fan\ tip}$ - r at stator tip, 0.427465 m

$r_{2,\,Stator\ tip}$ - r at stator tip, 0.427465 m

$r_{3,\,Stator\ tip}$ - r at stator tip, 0.427465 m

$r_{3,\,IGV\ tip}$ - r at IGV tip, 0.385615 m

$r_{4,\,IGV\ tip}$ - r at IGV tip, 0.385615 m

$r_{4,\,Rotor\ tip}$ - r at 1st HPC Rotor tip, 0.385615 m

$r_{5,\,Rotor\ tip}$ - r at 1st HPC Rotor tip, 0.385615 m

5.2 PARTICLE IMPACT AT 1ST HPC BLADE

5.2.1 DESIGN OF EXPERIMENTS

Experimentation will be run by the Design of Experiments methodology – under this systematic methodology the main factors that affect the outcome will be determined. The two particle sizes for experimentation, 1.33 and 3.2 mm diameter, were chosen because the public domain data on their velocity at impact has enough spread to cover all the impact velocities calculated by the In-Engine Kinetics Model (Section 4.8) – thus the public domain data on velocity to depth of penetration at impact can be leveraged to extract depths of penetration at the impact velocities calculated by the In-Engine Kinetics Model. Four input factors control the experimentation in the In-Engine Particle Kinetics model (Section 4.6): engine/ground clearance, stagnation point ground distance to fan face, inlet Mach number and Head Wind – the external geometry of the engine and the vortex ground conditions under the experimental conditions for this case study lead to strong vortices that will aspirate debris if it is located inside the ground vortex stagnation point (Sections 4.5 and 4.6). The experiment is run for each particle condition in the In-Engine Particle Kinetics model simulation (Appendix E) – this simulation code contains the inlet velocity and location of the ingested vortex at fan face fits from the Aspiration Model simulation run for the same experimental factors and levels as those of this case study, beta distributions for flow aero/thermos dynamics parameters from 10,000 runs (at 10,000 random inlet Mach numbers) of the Meanline Analysis model. The experimentation input factors are varied at three levels for a Full Factorial of 81 settings (see Table 5.1). To expand the coverage of the experiments and to consider internal experimental points 1,000 (points are evenly spaced) additional settings were concatenated to the Full Factorial 81 settings – these additional settings are called Space Filling Latin Hypercube designs, where levels are spaced evenly from lower bound to upper bound of each factor.

Table 5.1: Full Factorial Experimental Factor Settings – Particle Impact

Ground Clearance, h (m)		
0.24	0.285	0.33
Ground Distance (Ground Stagnation Point to Fan Face) (m)		
0.5	0.75	1
Inlet Mach Number (M_i)		
0.45	0.475	0.5
Wind Velocity (Head Wind, U_∞) (m/s)		
9.357	10.6785	12

5.2.2 EXPERIMENTATION

Before the raw data and trends for impacts at the rotor blades are discussed the qualitative nature of these will be discussed. The In-Engine Particle Kinetics code was developed with the capability to graph the engine internal geometry, impact events and particle trajectory. The code can also graph the airfoil geometry where impacts happen. As the code is computationally intensive (1,081 runs with full visuals takes over a week on 8 GB RAM laptop with an Intel® Core™ i5-3437U CPU @ 1.90 GZ) for full runs the full graphics routines are paused. Figures 5-2 and 5-3 present impact at the fan and exit out of 1st HPC rotor (no impact) respectively. Figures 5-3 and 5-5 reveal that the particle meanders slightly as it travels through the engine – this is consistent with the forces from tangential flows that are imposed on the particle (see Figure 4-36, Section 4.8.1). Figures 5-4 and 5-5 present impacts at the trailing edge and mid-section of the airfoil. Figure 5-6 reveals that axial particle velocity inside the engine has a slight increase as compared to its magnitude at ingestion (t =0) – this is consistent with the physics of the axial flow in the engine, the meridional flow velocity is larger than the initial particle velocity and thus provides an impulsive force to the particle (see the equations of motion in Section 4.8.1) – the particle does not outpace the flow impulsing it.

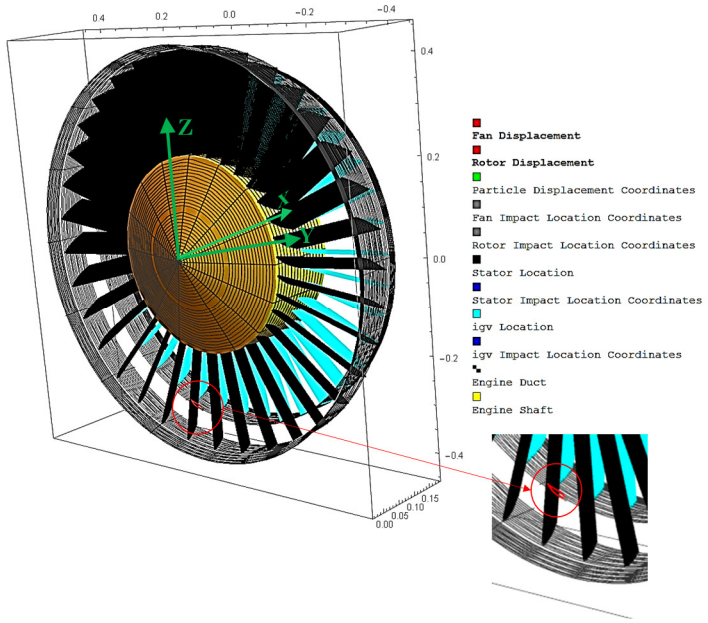


Figure 5.2: Particle Impact at Fan Blade

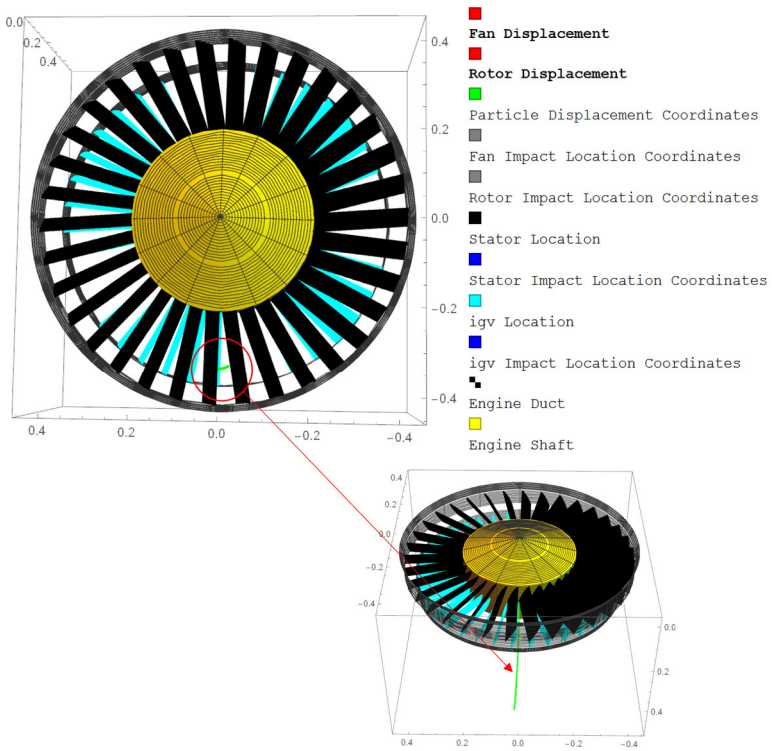


Figure 5.3: Particle Travels Out of 1st HPC Rotor Blade (no impact)

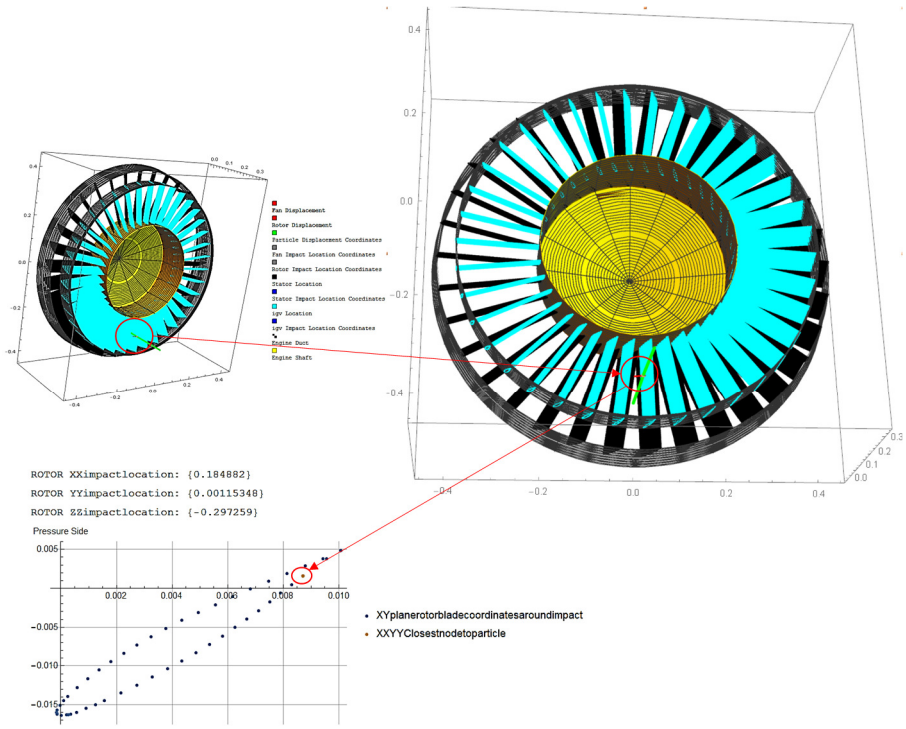


Figure 5.4: Particle Impact at 1st HPC Rotor Blade Trailing Edge

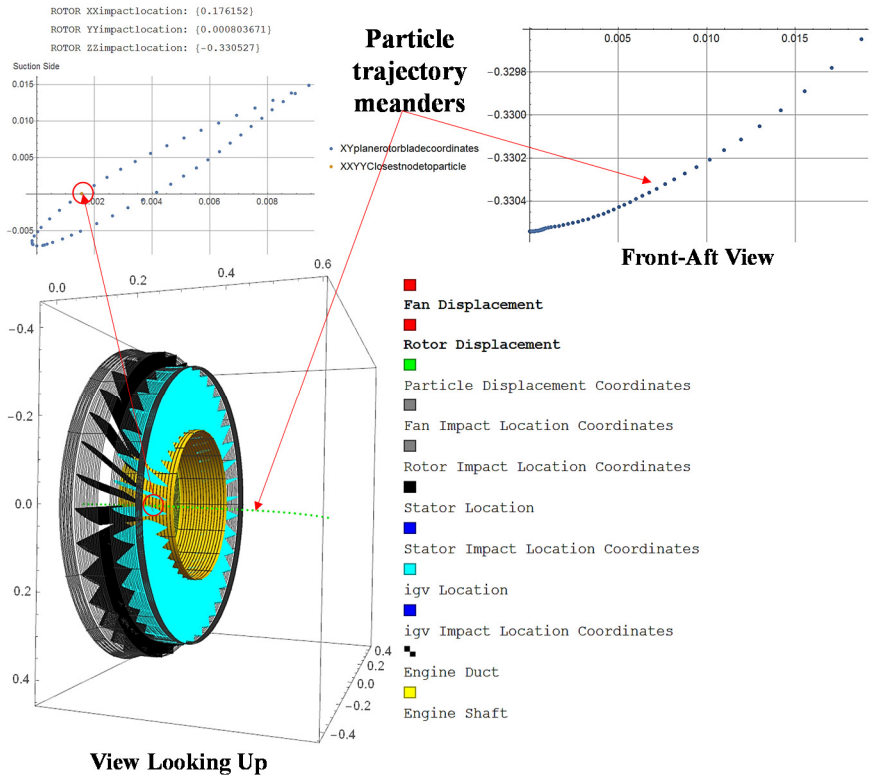


Figure 5.5: Particle Impact at 1st HPC Rotor Blade Mid-Section

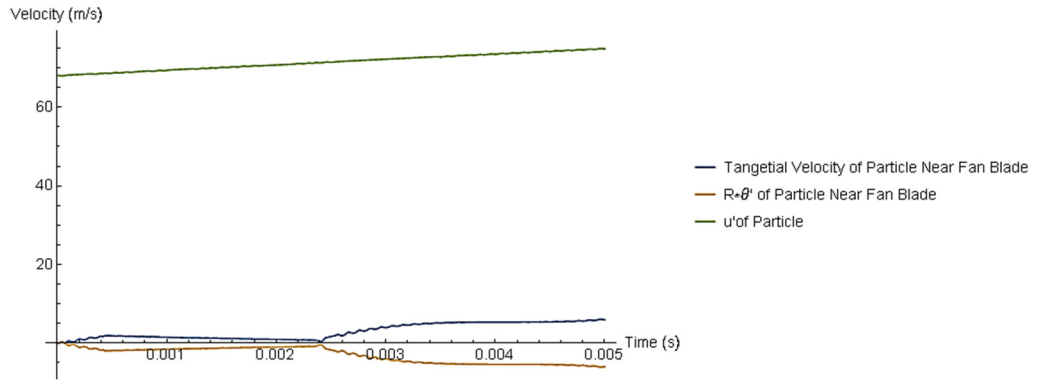


Figure 5.6: Axial and Tangential Particle Velocities

The radial and chord-wise locations of impacts on the blade reveal patterns that differ from the airfoil Suction Side to the Pressure Side. Figures 5-7 and 5-8 reveal that on the Suction Side impacts for the two particle sizes considered in this case study, 1.33 and 3.2 mm diameter, tend to congregate near the leading edge – the larger particle congregates more aggressively on the Leading Edge than the smaller one. Figures 5-7 and 5-8 also reveal that there are chord-wise gaps where no impacts occurred – this is a key finding since LSP treatments cannot be applied to the entire OML of the airfoil, area must be left untreated to carry left-over residual tensile stresses (Section 4.9.2.3). On the Pressure Side Figure 5-9 reveals that for the smaller particle the distribution of impacts is largely uniform – for the larger particle, radially on the blade from 0.33 to 0.34 m there are gaps where no impacts occurred (recall the blades extends from root to tip 0.21 to 0.385615 m) (see Figure 5-10). Figure 5-11 for the smaller particle on the Suction Side the higher impact velocities tend to congregate (chord-wise) near the leading edge of the airfoil – for the larger particle a more uniform distribution for velocity on the more narrow chord-wise locations of impact is observed (see Figure 5-12). On the Pressure Side of the airfoil for both particle sizes a "horseshoe" impact velocity vs. non-dimensional chord location distribution is revealed (see Figures 5-13 and 5-14). Recall that most tests that simulate particle impact on axial

compressor blades are performed on the leading edges of specimens [87, 88, 89, 90] – for the impact simulations on the two particles sizes considered in this case study the assertion of dominant leading edge impacts in the public domain literature is debunked. Though not a focus of this thesis, for three input settings (DOE settings) Figure 5-15 reveals that the planar trajectory of the particle has a "kink" – that "kink" suddenly ends when the particle enters the 1st HPC rotor where whirl velocity is significantly higher than on the fixed stators and IGVs – the higher whirl velocity angularly "straightens" the planar trajectory of the particle. This finding perhaps hints that if there had been more stationary stations between the fan and the 1st HPC rotor the particle may have spun inwards to the shaft. Figure 5-15 also evinces a less aggressive trajectory in the Y axis (measured from engine centerline) when the kink is more pronounced.

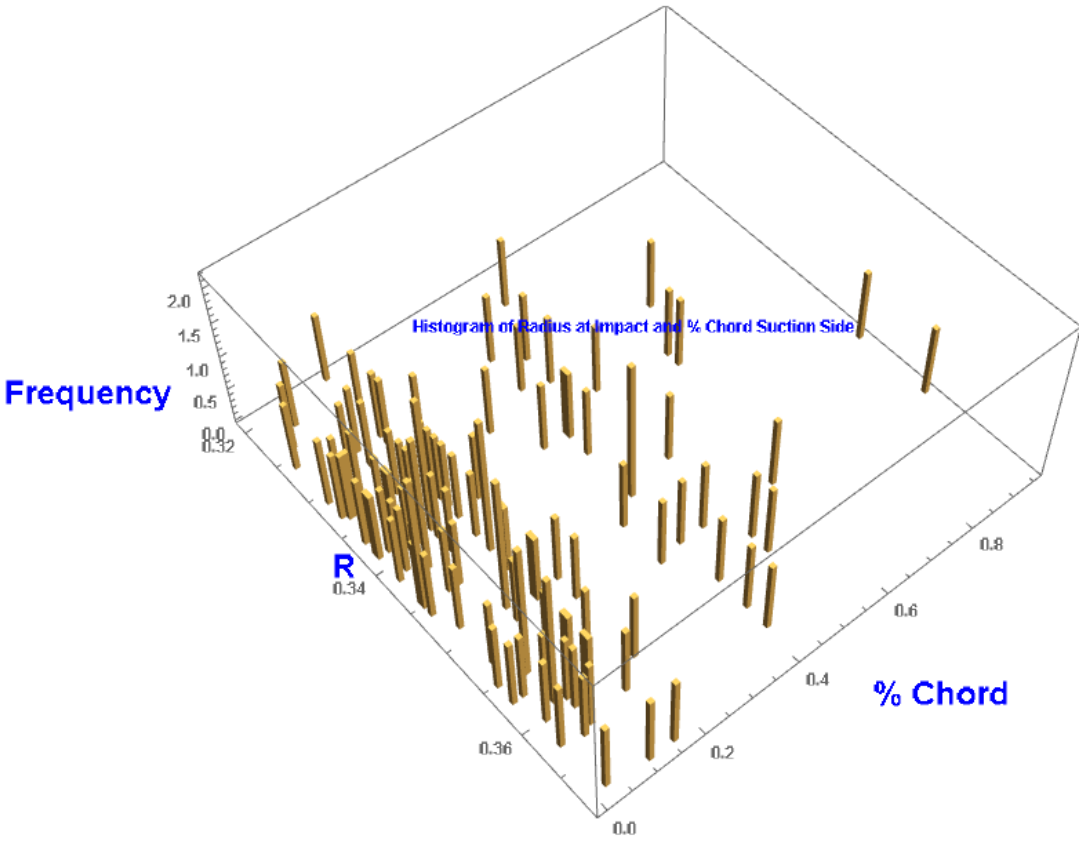


Figure 5.7: R vs. Non-dim Chord Impact Distribution, Suction Side, 1.33 mm FOD

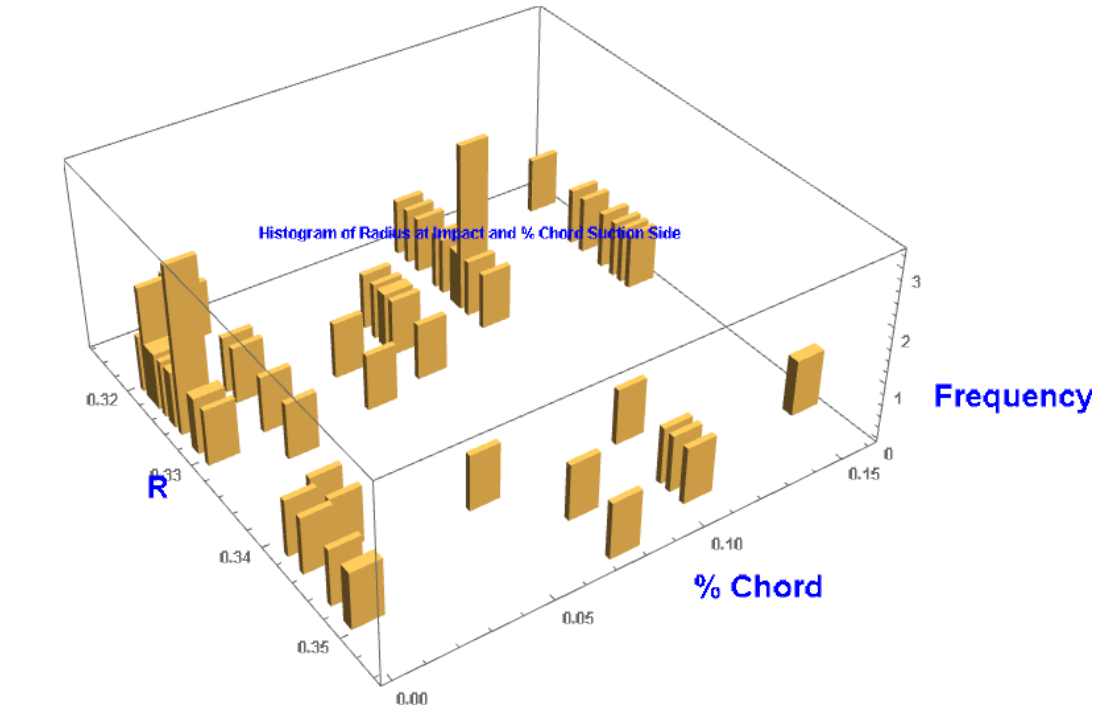


Figure 5.8: R vs. Non-dim Chord Impact Distribution, Suction Side, 3.2 mm FOD

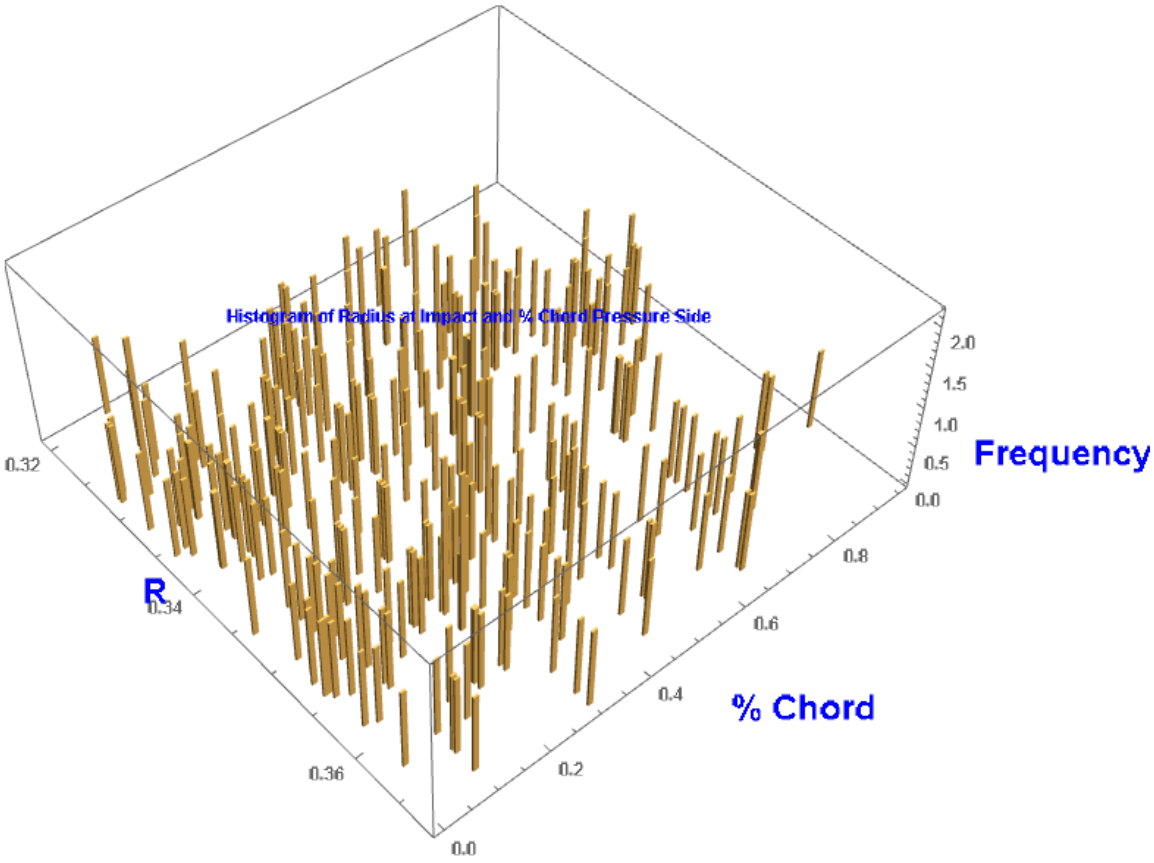


Figure 5.9: R vs. Non-dim Chord Impact Distribution, Pressure Side, 1.33 mm FOD

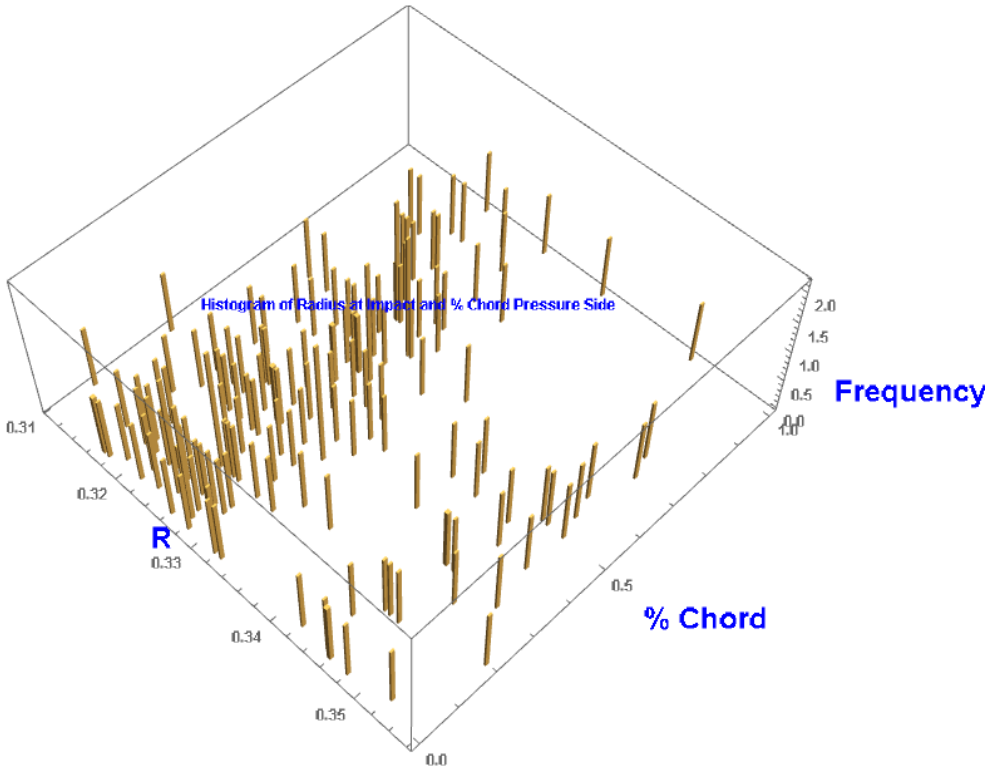


Figure 5.10: R vs. Non-dim Chord Impact Distribution, Pressure Side, 3.2 mm FOD

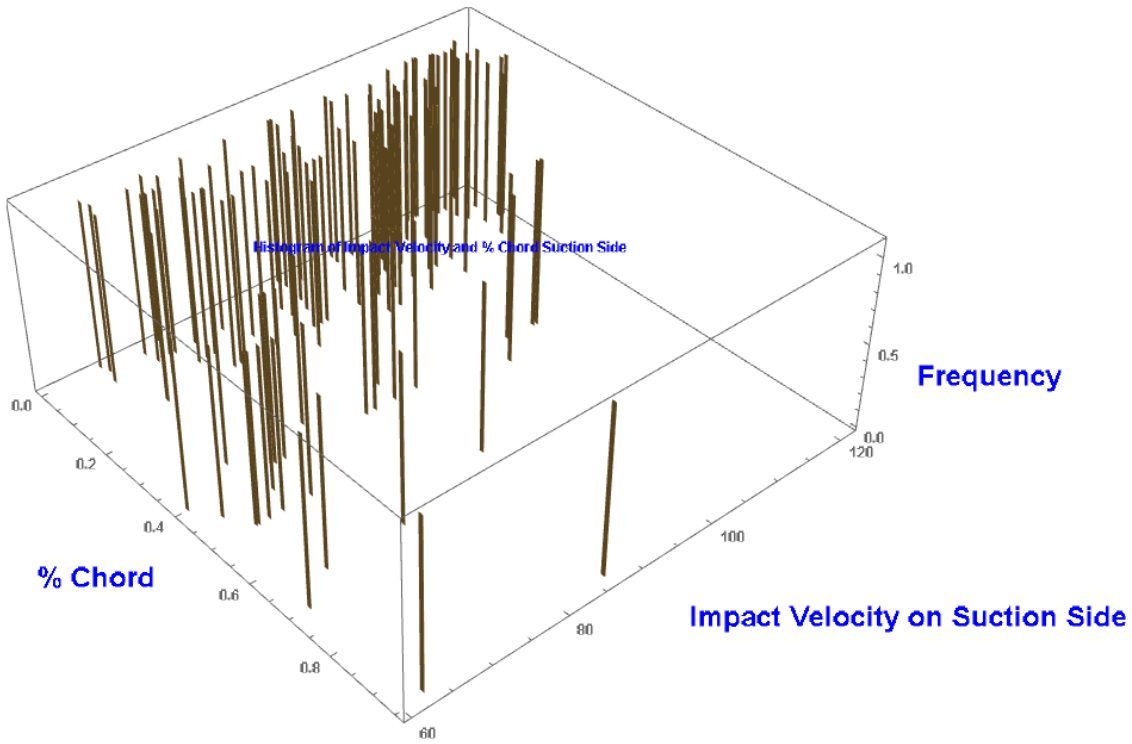


Figure 5.11: Non-dim Chord vs. Impact Velocity, Suction Side, 1.33 mm FOD

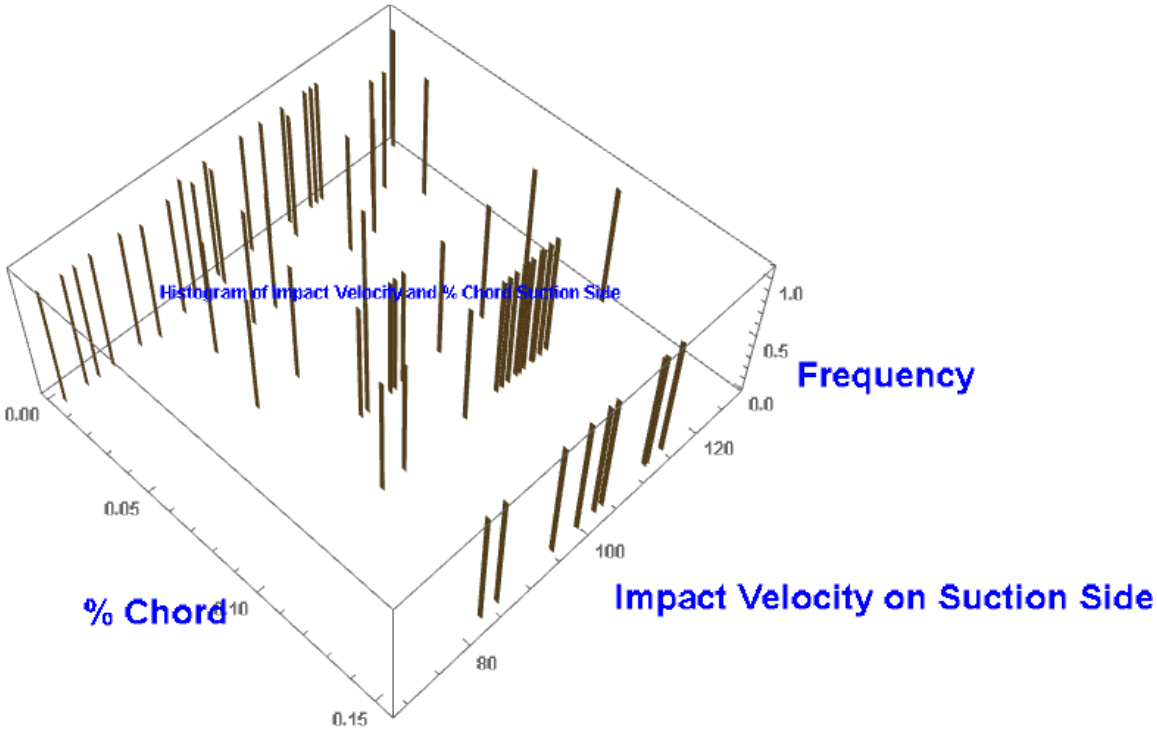


Figure 5.12: Non-dim Chord vs. Impact Velocity, Suction Side, 3.2 mm FOD

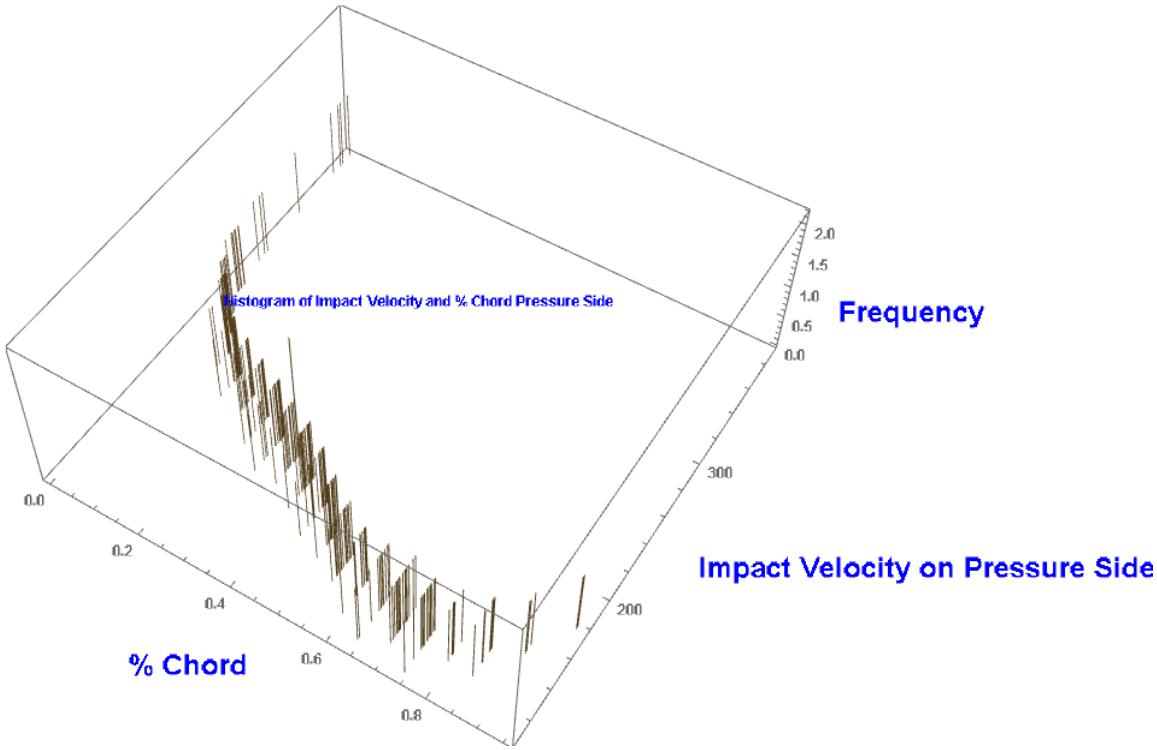


Figure 5.13: Non-dim Chord vs. Impact Velocity, Pressure Side, 1.33 mm FOD

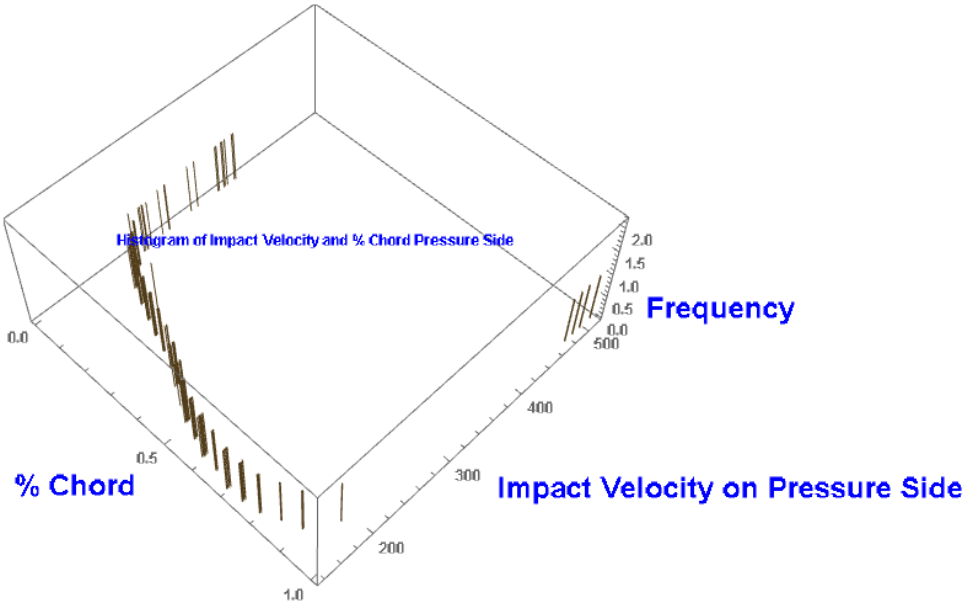


Figure 5.14: Non-dim Chord vs. Impact Velocity, Pressure Side, 3.2 mm FOD

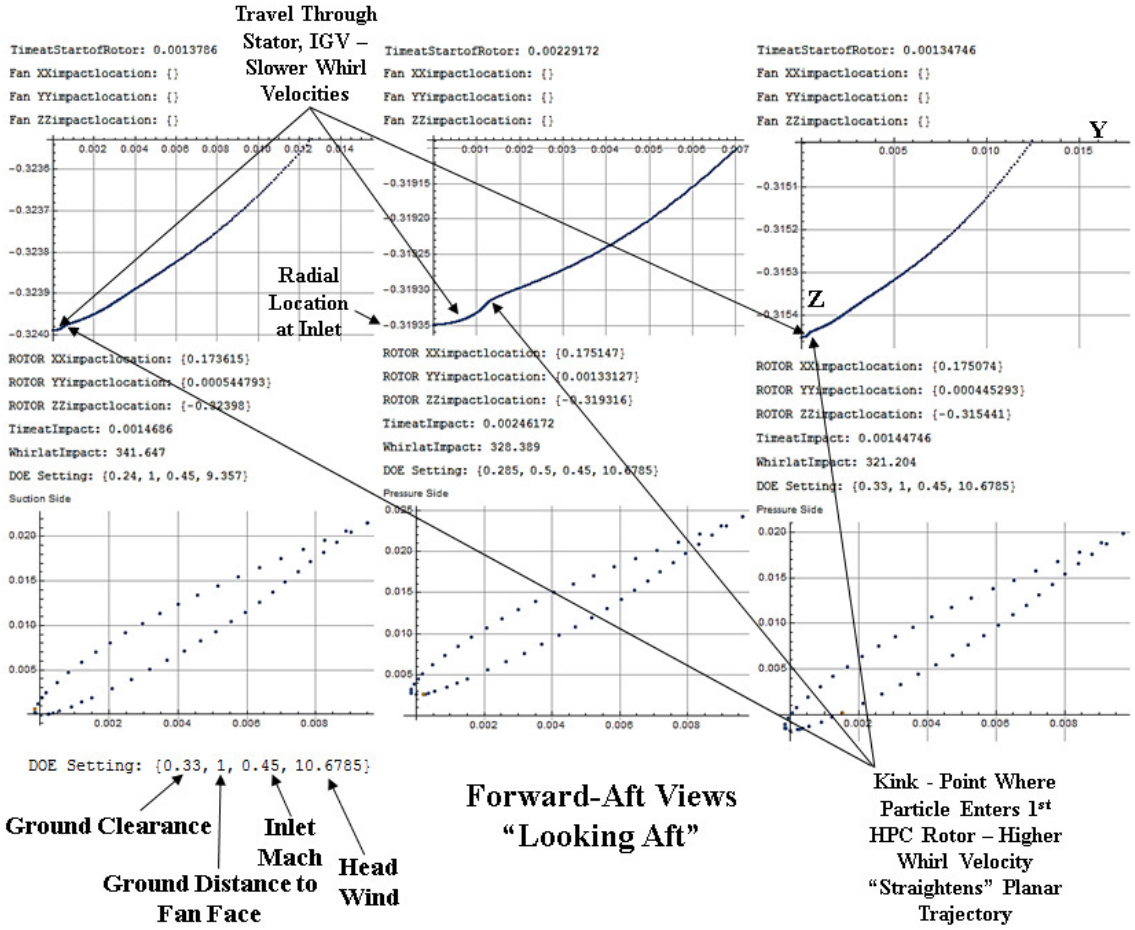


Figure 5.15: Planar Particle Trajectory, Forward-Aft View for 3 DOE Settings

Not all particles strike blades at the 1st HPC rotor – though only the 1.33 and 3.2 mm mild steel particles were fully studied for reasons already mentioned the In-Engine Particle Kinetics simulation was run an additional size, 2 mm diameter (see Table 5.2). The regime being considered for the case study is where Head Wind is equal to Rolling Ground – Murphy's regimes for Quiescent (no wind) and Boundary Layer (with Headwind but no Rolling Ground) were run in the In-Engine Particle Kinetics model (see Table 5.3). Tables 5.2 and 5.3 evince the following:

- As particles get larger fewer of them impact the 1st HPC rotor blades
- The airfoil Pressure Side impacted is more than twice that of the Suction Side
- A local boundary layer effect (local to Murphy's experiments, recall that vortex vertical location is extracted and corrected from his scaled model to the full size model of the case study, for details go to Section 4.5.3.2) lowers the location (height) where the vortex is being ingested at the inlet, resulting in a sizeable number of particles exiting via engine bypass section
- As particles get larger more of them impact the fan
- The impact rate (total impacts at 1st HPC rotor blades to simulation runs) drops significantly from particle diameter 2 mm to 3.2 mm (vs. a more gradual drop from particle diameter 1.33 mm to 2 mm)
- Impacts do not occur radially at all points nor at all non-dimensional chord locations on a blade (the blade has a shaft radius of 0.21 m and a tip radius of 0.385615 m) – impacts radially range from 0.2966 to 0.3855 m
- The data from Table 5.3 comprises the radial/chord-wise map where LSP treatments will be applied to the OML of the blade

Table 5.2: Particle Impact Data for Experimental & Two Additional Regimes

On Runway Regime	Particle Diameter (mm)	Runs	1st HPC Rotor Blade Total Impacts	Pressure Side 1st HPC Rotor Blade Total Impacts	Suction Side 1st HPC Rotor Blade Tota Impacts	Fan Impacts	Exited via Bypass	Impacted Engine Housing	Exited 1st HPC Rotor (no impact)	Impact Rate (1st HPC Rotor)
Quiescent (no wind)	1.33	842	293	198	95	246	0	0	302	0.348
$U_\infty = U_g$ (Head Wind = Rolling Ground)	1.33	1081	402	283	119	303	0	0	375	0.372
Boundary Layer, Headwind	1.33	1243	276	190	86	363	346	0	257	0.222
Quiescent (no wind)	2	778	268	180	88	Not recorded	Not recorded	Not recorded	N/A	0.344
$U_\infty = U_g$ (Head Wind = Rolling Ground)	2	1081	349	280	69	534	0	0	197	0.323
Boundary Layer, Headwind	2	1243	239	174	65	616	243	2	144	0.192
Quiescent (no wind)	3.2	934	181	129	52	655	0	0	97	0.194
$U_\infty = U_g$ (Head Wind = Rolling Ground)	3.2	1081	225	150	75	748	0	0	107	0.208
Boundary Layer, Headwind	3.2	1243	145	96	49	890	118	1	89	0.117

Table 5.3: Impact Locations

			Radial Impact Location (m)		Non-dim Chord (from Leading Edge, m)	
Impact Side	**Particle Diameter (mm)**	**On Runway Regime**	**Min**	**Max**	**Min**	**Max**
Suction	1.33	Quiescent (no wind)	0.2966	0.3427	0	0.7507
Suction	1.33	$U_\infty = U_g$ (Head Wind = Rolling Ground)	0.3197	0.3683	0	0.7014
Suction	1.33	Boundary Layer (BL), Headwind	0.324	0.3852	0	0.7999
Suction	2	Quiescent	0.2966	0.3427	0	0.7507
Suction	2	$U_\infty = U_g$	0.316	0.3591	0	0.2026
Suction	2	BL	0.3259	0.384	0	0.2529
Suction	3.2	Quiescent	0.3024	0.3371	0	0.1521
Suction	3.2	$U_\infty = U_g$	0.3197	0.3273	0	0.3495
Suction	3.2	BL	0.3269	0.3854	0	0.3853
Pressure	1.33	Quiescent	0.2966	0.3417	0.005	0.8459
Pressure	1.33	$U_\infty = U_g$	0.3197	0.3677	0.005	0.9464
Pressure	1.33	BL	0.3242	0.3856	0.005	0.9464
Pressure	2	Quiescent	0.2966	0.3417	0.005	0.8459
Pressure	2	$U_\infty = U_g$	0.312	0.359	0.005	0.9963
Pressure	2	BL	0.3259	0.3855	0.005	0.9963
Pressure	3.2	Quiescent	0.3	0.3377	0.005	0.9963
Pressure	3.2	$U_\infty = U_g$	0.3109	0.3555	0.005	0.9963
Pressure	3.2	BL	0.3265	0.3855	0.005	0.9963

5.2.3 DOMINANT CONTRIBUTORS

The research question still to be considered is which experimental factors (recall factors in the DOE are engine/ground clearance, stagnation point ground distance to fan face, inlet Mach number and Head Wind) contribute most to the outputs recorded by the

simulation code – the full set of recorded outputs are impact occurrence at a 1st HPC rotor blade (does it take place or not), if an impact occurs the radial location, non-dimensional chord location, impact side, impact velocity, axial stress at radial location, depth of penetration, width of penetration, chord length, impact node, local thickness and geometry factors (Betas for Stress Intensity). Though recorded, the entire set of outputs is not required to gage the importance of the experimental factors because some outputs are indexed to others – employing the simulation code notation StressatImpactSite is a function of radial location Rimpact via Equation 165, width of penetration DiameterofPenetration is a function of DepthofPenetration via Equation 181, chord length at impact ChordatImpact is function of Rimpact via Equation 152b, the impacted node on the airfoil Nimpact is determined in the simulation code via the impact law where the node with the smallest distance that is less than or equal to r_{FOD} to the center of the particle becomes the impacted node (recall that as the NACA 65-210 airfoil is discretized into 51 nodes, non-dimensional chord location along with impact side are indexed to the impacted nodes) (Section 4.8.2.6), LocalThicknessatImpact is calculated in the simulation code as the distance from the impacted node to the node directly opposite (refer to code in Appendix F for routine).The geometry factors $\beta_{compounded}$ are not necessary as the Crack Growth and LCC model computes them. Thus Rimpact, Xnondimchord, ImpactVelocity, ImpactSide and DepthofPenetration become the outcomes to which the contribution of the experimental factors will be gaged – LocalThicknessatImpact is indexed to non-dimensional chord location it need not be analyzed as it will provide the same trending information. The experimental factors from the DOE for both particle sizes and associated outcomes were converted to a data table in SAS Statistical Discovery Software JMP® and

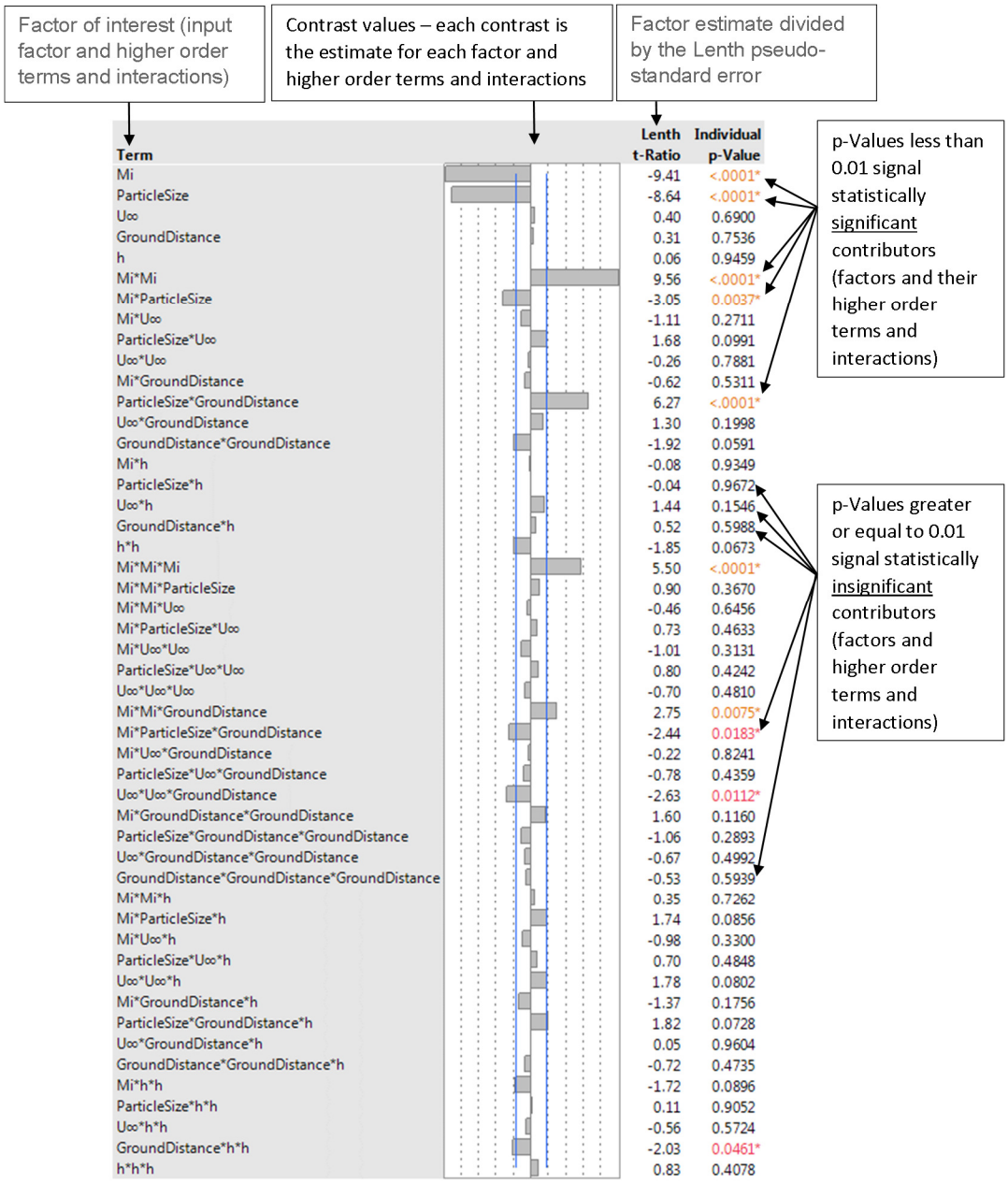
a screening analysis was ran (screening tests identify which factors have the greatest effect on the outcomes). Additionally, histograms plots of the factor settings vs. output were generated to support the screening test.

5.2.3.1 Contributors & Trends for Impact Occurrence

Figures 5-16 presents the statistically significant factors for the output (impact occurrence recorded in binary mode, 1 for impact and 0 for no impact), whereas Figure 5-17 presents how via a Histogram plot trending information can be discerned – both figures can be used by the reader as rubrics to discern significant contributors and trends as they present means to infer or distinguish tendencies. The salient findings are:

- Figure 5-16 reveals that for the range of settings chosen the factors and their higher order terms and interactions that dominate whether a particle impacts a blade on the 1st HPC rotor or not are: M_i, ParticleSize, M_i*M_i,M_i*ParticleSize, ParticleSize*GroundDistance, M_i*M_i*M_i and M_i*GroundDistance (assuming a p-Value ≤ 0.01 indicates strong evidence against the null hypothesis that the factors are not statistically significant)
- Figure 5-17 corroborates that particle size and inlet Mach number contribute most to whether a particle will impact a rotor blade - the larger particle in the larger inlet Mach number range, in the smaller engine/ground clearance range and in the smaller ground distance from the stagnation point to the fan face range has the largest frequency of non-impacts
- The largest frequency of impacts is had from the smaller particle in the smaller inlet Mach number range, in the smaller engine/ground clearance range and in the larger ground distance from the stagnation point to the fan face range - as

the ground distance from the stagnation point to the fan face increases so does particle velocity at inlet and because the flow meridional velocity does not change appreciably across the engine stations (see Section 4.7) higher particle velocities inside the engine occur, this relationship along with a higher rate of impact at the Fan station by the larger particle (refer to Table 5.2) corroborates that the smaller particle that moves more quickly inside the engine will impact the 1st HPC rotor blades more often – the smaller, faster moving particle more often misses the slower rotating fan (Ω of 10,000 rpm at fan vs 14,000 rpm at HPC rotor) and reaches the faster rotating HPC rotor where it has a greater rate of impact and of passing through that station with no impact than the larger particle (refer to Table 5.2)



Term	Lenth t-Ratio	Individual p-Value
Mi	-9.41	<.0001*
ParticleSize	-8.64	<.0001*
U∞	0.40	0.6900
GroundDistance	0.31	0.7536
h	0.06	0.9459
Mi*Mi	9.56	<.0001*
Mi*ParticleSize	-3.05	0.0037*
Mi*U∞	-1.11	0.2711
ParticleSize*U∞	1.68	0.0991
U∞*U∞	-0.26	0.7881
Mi*GroundDistance	-0.62	0.5311
ParticleSize*GroundDistance	6.27	<.0001*
U∞*GroundDistance	1.30	0.1998
GroundDistance*GroundDistance	-1.92	0.0591
Mi*h	-0.08	0.9349
ParticleSize*h	-0.04	0.9672
U∞*h	1.44	0.1546
GroundDistance*h	0.52	0.5988
h*h	-1.85	0.0673
Mi*Mi*Mi	5.50	<.0001*
Mi*Mi*ParticleSize	0.90	0.3670
Mi*Mi*U∞	-0.46	0.6456
Mi*ParticleSize*U∞	0.73	0.4633
Mi*U∞*U∞	-1.01	0.3131
ParticleSize*U∞*U∞	0.80	0.4242
U∞*U∞*U∞	-0.70	0.4810
Mi*Mi*GroundDistance	2.75	0.0075*
Mi*ParticleSize*GroundDistance	-2.44	0.0183*
Mi*U∞*GroundDistance	-0.22	0.8241
ParticleSize*U∞*GroundDistance	-0.78	0.4359
U∞*U∞*GroundDistance	-2.63	0.0112*
Mi*GroundDistance*GroundDistance	1.60	0.1160
ParticleSize*GroundDistance*GroundDistance	-1.06	0.2893
U∞*GroundDistance*GroundDistance	-0.67	0.4992
GroundDistance*GroundDistance*GroundDistance	-0.53	0.5939
Mi*Mi*h	0.35	0.7262
Mi*ParticleSize*h	1.74	0.0856
Mi*U∞*h	-0.98	0.3300
ParticleSize*U∞*h	0.70	0.4848
U∞*U∞*h	1.78	0.0802
Mi*GroundDistance*h	-1.37	0.1756
ParticleSize*GroundDistance*h	1.82	0.0728
U∞*GroundDistance*h	0.05	0.9604
GroundDistance*GroundDistance*h	-0.72	0.4735
Mi*h*h	-1.72	0.0896
ParticleSize*h*h	0.11	0.9052
U∞*h*h	-0.56	0.5724
GroundDistance*h*h	-2.03	0.0461*
h*h*h	0.83	0.4078

Figure 5.16: Screening Test, Input Factors vs. Impact Occurrence

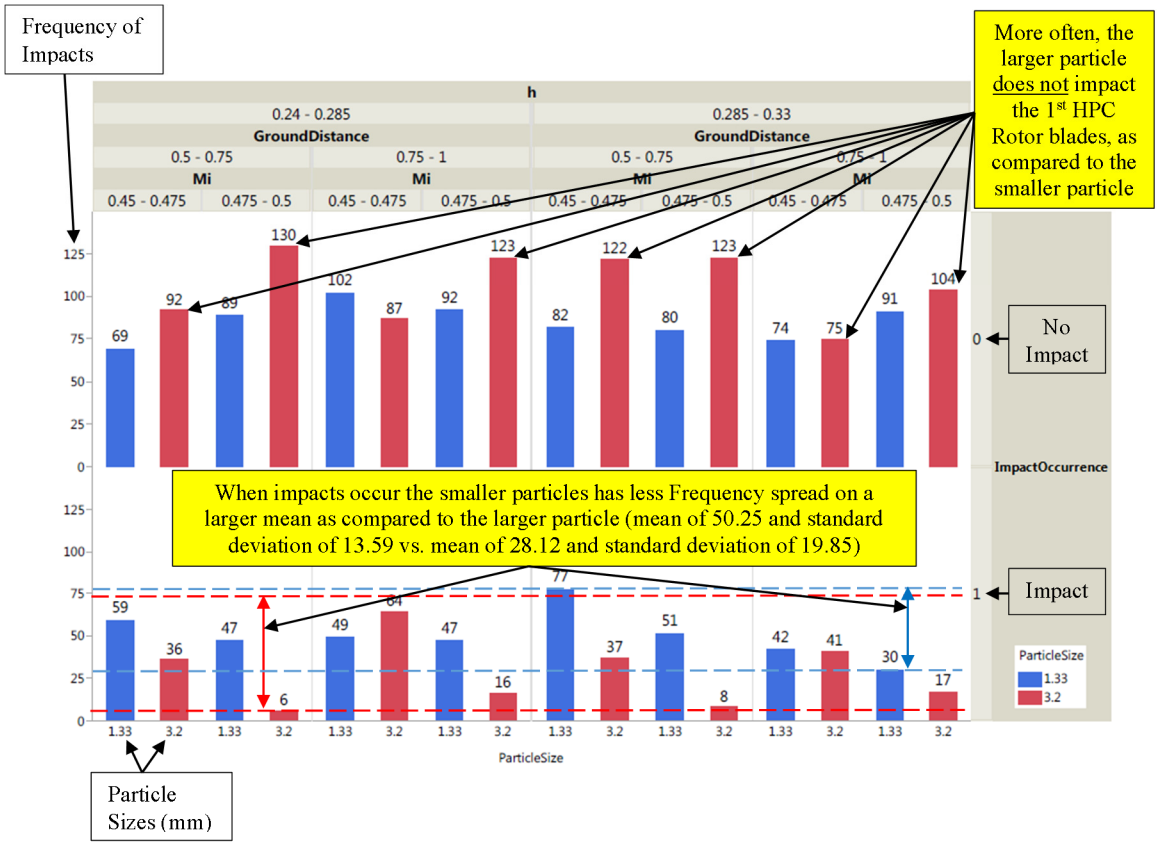


Figure 5.17: Histogram, Input Factors vs. Impact Occurrence

5.2.3.2 Contributors & Trends Once Impact Occurs

The subset of data once particles impact the blades is run through the screening analysis. For the screening test plots for radial location, non-dimensional chord-wise location on the airfoil, impact side, impact velocity and depth of penetration of a particle that does impact a 1st HPC rotor blade refer to Appendix H employing Figure 5-16 as rubric to discern statistically significant factors. Figures 5-18 presents trends for the radial location of where an impact occurs on a rotor blade for the dominant factors (particle size, inlet Mach number, ground clearance to inlet and Head Wind) and their higher order terms and interactions. Likewise Figure 5-19 presents trends from the data points for the non-dimensional chord location, velocity and depth of penetration of where an impact occurs

on a rotor blade for the dominant factors (particle size and ground distance from stagnation point to fan face) and any higher order terms and interactions. The salient findings are:

- More impacts occur on the pressure side of the airfoil for the smaller particle (also evident in Figures 5-7 through 5-10)
- Figure 5-18 reveals that particle size is most consequential for radial location of impact – the larger particle impacts closer to the engine centerline (all other factors held constant) – it has greater momentum than the smaller particle that is more susceptible to being spun out by the flow whirl component
- Also revealed from Figure 5-18, inlet Mach number and inlet ground clearance have a sizeable effect on radial location of impact – as inlet Mach number increases particles strike further from the engine centerline axis and as the engine moves higher up particles impact closer to the engine centerline axis – both of these trends are consistent with Murphy's experimental data of the location of the vortices at the inlet and the trajectories for the particle sizes chosen for the case study do not meander significantly (axial momentum dominates over flow whirl component effects on the particles) – this consistency is validation that for impact locations the particle kinetics model provides useful trending information [96] (refer to Section 4.5.3 and Appendix B) – recall that vertical location of the particle at the inlet is related to radial location at the start of the simulation by R = Dl/2+h-Z where the vertical location Z is calculated from the fit in Appendix B (corrected vortex vertical location at inlet) that is a function of h/Dl, non-dim boundary layer, inlet Mach number, Head Wind ($U\infty$) and Rolling Ground - for this case study there is zero

boundary layer and Rolling Ground is equal to Head Wind, h is one of the factors and D_l is fixed

- The screening test plots in Appendix H reveal that particle size and stagnation point to fan face ground distance are statistically significant for non-dimensional chord location, impact velocity and depth of penetration at the point of impact – this finding is intuitive for impact velocity (also intuitive for depth of penetration as they are statistically indexed) as the longer the ground distance from the stagnation point to the fan face the more time the particle has to convert more of its velocity to Vx (refer to Section 4.6.3 for more detail) - none of the factors are statistically significant to impact side (assuming a p-Value ≤ 0.01 indicates strong evidence against the null hypothesis that the factors are not statistically significant)
- Figure 5-19 reveals a weak trend for non-dimensional chord location (employing Figure 5-18 as a rubric to discern trends) – exhibited only for the smaller particle as the ground distance from the stagnation point to the fan face is increased the strikes occur closer to the leading edge of the airfoil - this is intuitive as increasing ground distance increases particle velocity at inlet, the impacts that occur tend to be toward the leading edge, otherwise the particle may pass through without impact because the NACA 65-210 airfoil quickly tapers aft
- Figure 5-19 also reveals that ground distance from stagnation point to the fan face has a noticeable effect on impact velocity - as ground distance increases so does impact velocity, in linear fashion - this is intuitive as increasing ground

distance increases particle inlet velocity and that velocity has a slight increase in the in-engine trajectory to the rotor blades - the same trend is present for depth of penetration (depth of penetration is linked to impact velocity by the statistics presented in Table 4.6 and the process in Figure 4-45)

- Upon impact the smaller particle causes smaller penetration depths – this finding, evident in Figure 5-19, along with the discovery that the mode of impact is different if it occurs on the pressure or suction side (on the pressure side of the airfoil the impact velocity is more severe because the event is similar to a "paddle" strike with a very fast rotating rotor striking the particle, whereas on the suction side it is the slower velocity at impact due to the particle striking the blade that occurs) address and support **Research Question 3** and **Hypothesis 3** respectively in this thesis – indeed there are factors that influence LCC affordability more significantly vs. other factors that were also varied during the experiments: the smaller particle impacts the rotor blades more often, but their depths of penetration are smaller, imparting stress intensities (via Beta geometry factors that are more benign) at the point of impact that will grow cracks more slowly, causing less damage that leads to fewer blade replacements and reduced LCCs (see Section 5.3)

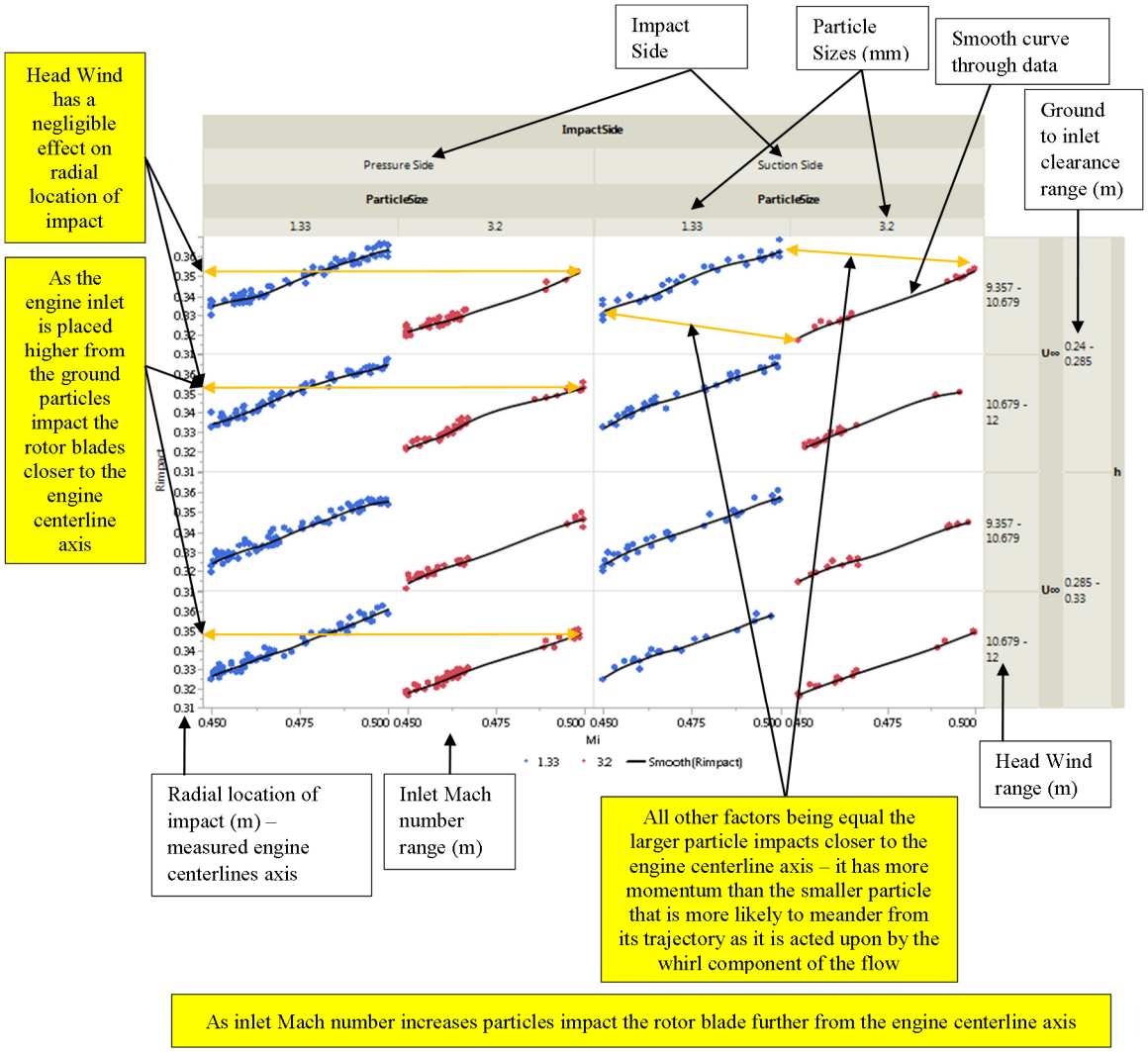


Figure 5.18: Data Trends, Radial Impact Location vs. Factors

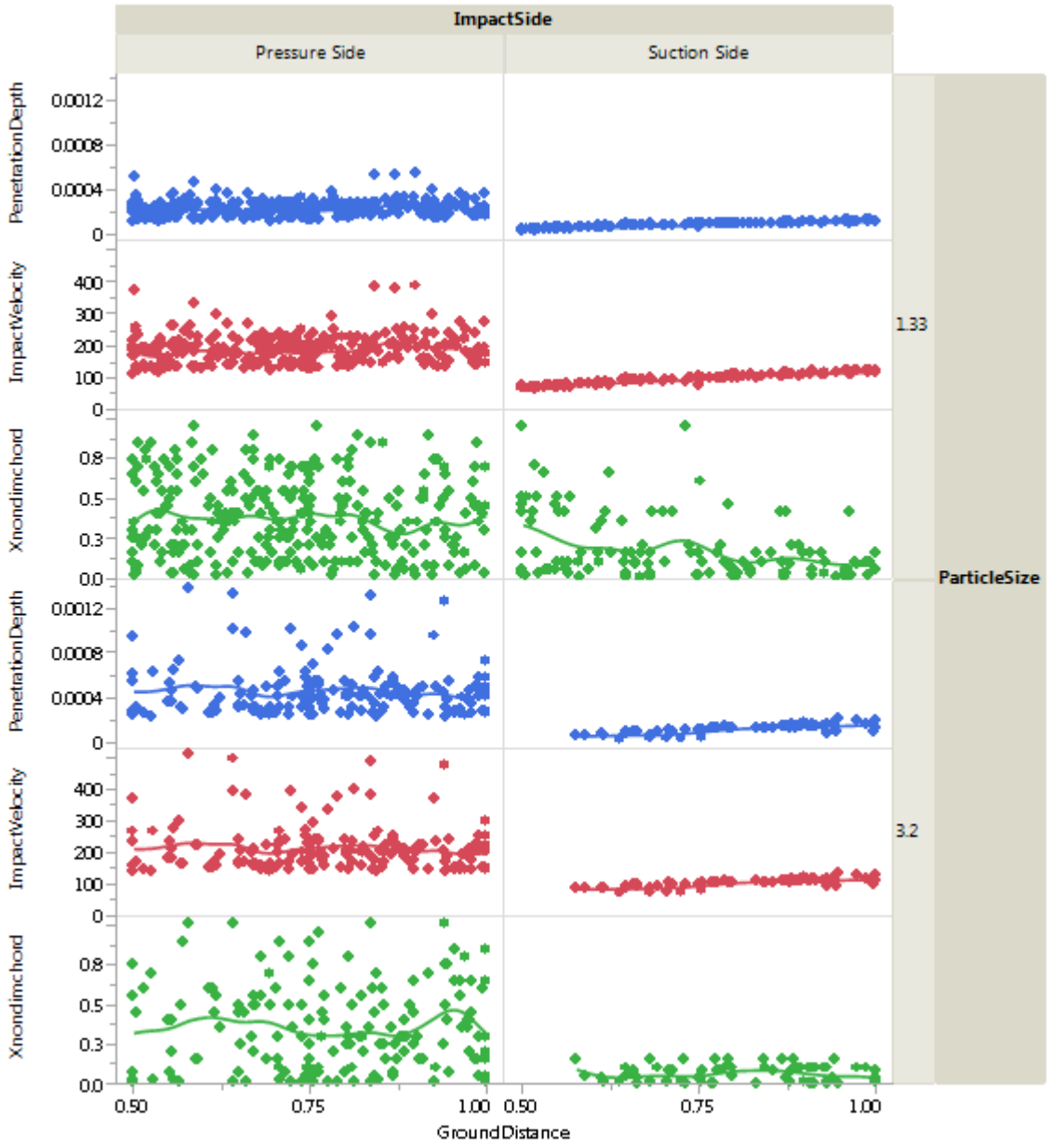


Figure 5.19: Data Trends, Non-Dim Chord, Impact Velocity & Depth vs. Factors

5.3 REPAIRS, REPLACEMENTS AND LCCS DUE TO IMPACTS

5.3.1 DESIGN OF EXPERIMENTS

From the experiments for particle impact at the 1st HPC rotor blades probabilities for impact were estimated by the ratio of number of impacts to runs (for the condition considered for this case study for each particle size a total of 1,081 simulations were ran).

This probability that the particle would impact either the Pressure or Suction Side was estimated from the ratio of the total number either side was recorded to the total number of impacts for each particle size. The probability that a particle is engulfed by the core radius of the ground vortex stagnation point if calculated from Equations 27a or 27b for each particle size considered for this case study. The Crack Growth and LCC simulation code contains these probabilities and their application in the architecture of the software (refer to Appendix F). Recall from the previous experiment that engine/ground clearance, stagnation point ground distance to fan face, inlet Mach number and Head Wind varied according to a DOE that contained 81 Full Factorial and 1,000 Space Filling Latin Hypercube experimental settings (Section 5.2.1). For the experiments that will shed light into how many repairs or replacements and their associated LCCs occur because of FOD particle impact a different set of factors will be considered - the four experimental factors of the previous experiment will be allowed to vary uniformly in the same ranges (but not by DOE) as they will be the inputs for the fits and distributions that were made for the outputs of the particle impact experimentation (radial impact location, non-dimensional chord location, impact velocity, impact side, depth of penetration, local impact thickness), their fits with goodness of fit measures and distributions are included in Appendix G for both sizes. The crack growth and LCC simulations for Visual and Automated inspections of the runway share four experimental factors: particle diameter, runway sweep (sweeping by towed sweeper), LSP and DTA engine inspections (interval engine inspections). In addition to these four the crack growth and LCC simulation that considers Visual Inspections of the runway for debris depends on one more experimental factor, the number of visual inspections performed in an operational, 24 hour day. The simulation considers

two additional experimental factors for Automated runway debris inspections, scan rate and probability of detection. Sections 4.9 and 4.10 provide extensive documentation for how crack growth and LCCs are inter-related for both FOD particle sizes and both runway inspection methods. The experimental factor setting ranges for both FOD particle sizes are presented in Table 5.4 – for the experiments that consider Visual runway inspections the Full Factorial is 24, whereas for the Automated runway inspections it is 72.

Table 5.4: Experimental Factor Settings – Crack Growth & LCCs

Runway Inspection Method					
Visual			Automated		
Number of Runway Visual Inspections			**Scan Rate**		
2	3	4	0.33	0.50	1.00
Number of Runway Sweeps			**PAutomatedDetection**		
1	2		0.65	0.695	0.74
LSP			Number of Runway Sweeps		
0	2		1	2	
DTA Engine Inspections			LSP		
0	1		0	3	
			DTA Engine Inspections		
			0	1	

5.3.2 EXPERIMENTATION

The experiments for crack growth and LCC estimation are less computationally costly as those for particle impact simulation and they are subject to the variability of all the model fits and distributions that contribute to it (uniform, random cost models for acquisition of LSP and Automated runway inspection equipment, uniform, random models for process, clean up and change over times for LSP treatment, uniform, random models for probability of aspiration, rotor strike and airfoil impact side (Pressure or Suction side), uniform, random models for variation of input (engine ground clearance, ground distance

from stagnation point to fan face, inlet Mach number and Head Wind) for fits and distributions of outputs from particle impact simulation (axial stress at impact, chord length at impact, radial location at impact, local thickness at impact, non-dim chord location on airfoil (measured from leading edge) and depth of penetration). The decision was made to perform replicates on these experiments to improve the precision of the results, to detect smaller effects or interactions that with fewer test runs would not be captured. Recall that the simulations for particle impact and subsequent crack growth focus on a single engine, while the LCCs are scaled to fleet level – the scaling to fleet level is accomplished by accounting for the size of the fleet, engines per aircraft and HPC rotors per engine. A statistics bootstrap plot of the standard deviation of the mean LCCs for the fleet vs. number of replicates of the experiment was employed to determine the appropriate number of experimental replicates – the number at which the standard deviation of the mean LCCs no longer varied significantly (<10%). From Figure 5-20 it can be determined that 100 replicates is the number.

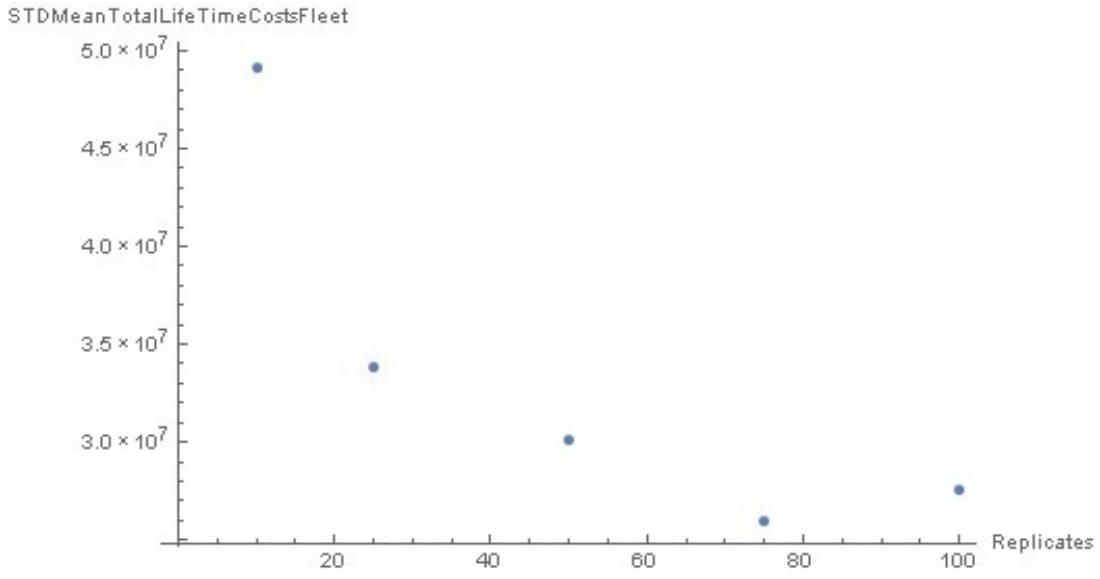


Figure 5.20: "Bootstrap" of Standard Deviation of Mean LCCs vs. Replicates

The questions should now be: what is the process described perhaps by pseudo code for this large simulation (the simulation code for Crack Growth and LCC estimation resides in Appendix F)?, how are the experiments run?. Figure 5-21 presents the process by which the crack growth and LCC simulation is run.

```
Do[
   Do[
      Extract DOE Experimental Settings for Visual or Automated Inspections;
      Calculate cost of inspecting and sweeping runway;
      ProbAspiration = calculate probability that particle lies inside on -
                       ground vortex core;
      If[RandNumber ≤ ProbAspiration,
         Aspiration triggered;
         ProbRotorStrike = # Rotor strikes at each FOD particle
                           size / experiment runs (DOE length);
         If[RandNumber ≤ ProbRotorStrike,
            Strike triggered;
            Seed random numbers to generate random input variables for
            fits and distributions for Particle Impact;
            If[Random number ≤ Impacts at Pressure sides / Total Impact,
               Pressure Side, Suction Side];
            Initiate variables;
            Simulate takeoffs;
            If[Impact at Airfoil mid - section,
              Grow cracks;
              Check for LRU repair or replacements or engine replacements;
              Calculate LCCs (aggregate of runway visual inspections and sweeping,
              equipment acquisition, interval engine inspections,
              blade surface treatments, efficiency degradation,
              LRU repairs or replacements or engine replacements);
              Append Repairs, Replacements and LCCs,
              If[Impact at Airfil Leading or Trailing Edge,
                Grow cracks;
                Check for LRU repair or replacements or engine replacements;
                Calculate LCCs (aggregate of runway visual inspections and sweeping,
                equipment acquisition, interval engine inspections,
                blade surface treatments, efficiency degradation,
                LRU repairs or replacements or engine replacements);
                Append Repairs, Replacements and LCCs]]], {n, 1, 1 + Nreplicates}],
   {ii, 1, DOElength}]
```

Figure 5.21: Process/Pseudo Code for Crack Growth & LCC Simulation

The simulation displays a crack growth curve only if there is an impact at 1st HPC rotor blade. As impact likelihood and impact location are randomly uniform the simulation

code was engineered to allow any number of impacts at any location on a blade – recall that the 1st HPC rotor has 38 blades that are spinning at 14,000 rpm at the time of takeoff (when the risk of FOD ingestion is highest) it is unlikely that an impact will happen on top of the same blade at the same location, this is corroborated by the impact map for LSP treatment locations presented in Section 4.9.2.4 where the frequency for hundreds of impact locations is typically 1 – in the unlikely event that a double strike would occur the code checks that the induced impact depth for this new FOD particle is larger than the crack that was growing from the first impact, if so the growth curve is bumped up by the new flaw size. For visual or automated inspections the mode of growth is the same – what changes is the impact to LCCs as visual and automated runway inspection methods have differing costs and may reduce the likelihood of having an impact event. Thus experimentally for crack growth what is of value is the size of the particle, whether LSP or interval engine inspection will be implemented.

Eight sample outputs of crack growth for the two particle sizes where extracted to provide a visual format to gage the effectivity of LSP treatments and the technology of DTA interval engine inspections. In Figures 5-22 and 5-25 a single impact is presented along with the natural crack growth curve (crack growth if no impact had taken place at the impact site) – the curve below the first two is an overlay of the impact growth and natural growth – it is intended to visually demonstrate the benefit of LSP, DTA interval engine inspections and the effect of particle size. Figure 5-23 evinces the code's capability to capture multiple impacts and multiple benefits from treatments or interval inspections – in this case one crack is "caught" at the 1st interval engine inspection and repaired, while the second one is "caught" at the second interval inspection and likewise repaired – the net

effect is that the structural risk of the engine is reduced and the blade can last its intended design life. If DTA engine interval inspections are not implemented in the experimental setting (factor setting or run) then repairs would only be performed at the depot level teardown inspection at the end of the engine lifetime – the simulation records zero repairs and associated costs – this is a critical simulation event because as the paradigm stands the aircraft are only inspected at the end of the stipulated engine lifetime. Figure 5-24 provides contrast between a mid-section and an edge flaw – as the edge flaws are part-through cracks their growth is more aggressive and this is captured by the model – for this particular case LSP and DTA engine interval inspections were not implemented, a small impact depth ensured that growth would not be appreciably more severe than natural growth. It is for larger particles that the crack growth methodology developed in detail in Section 4.9 that pays dividends – Figures 5-26 and 5-27 present a very active capability of the code to capture complex, multi-impact and repair or replacement of LRU activity- analysis of the visual output reveals that for the larger particle that activity for replacement or repair of blades occurs more readily and growth is more severe. Figure 5-27 captures an even more complex scenario: the aggregate benefit of LSP surface treatment and DTA engine interval inspections – after multiple impacts, two interval inspections and an LRU replacement from FOD impacts the mid-section and edges of two blades the second blade is able to be "lifed" or extended up to its intended design life.

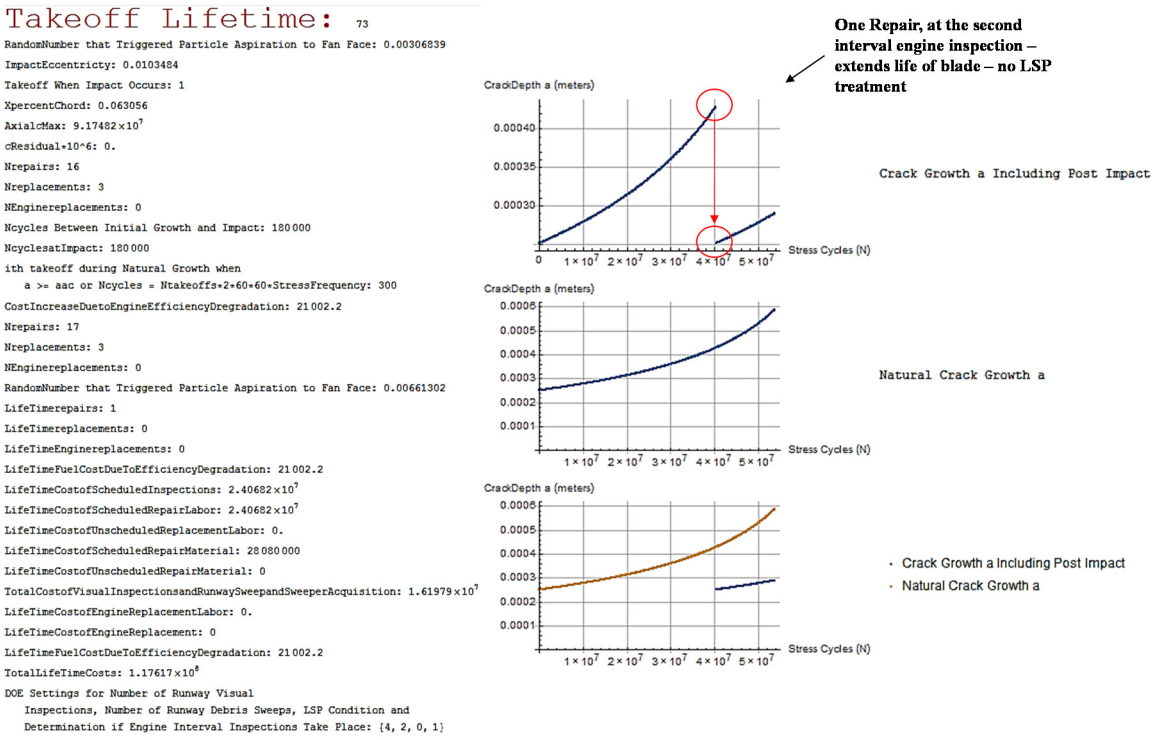


Figure 5.22: DTA Inspect. On, LSP Off, Single Impact, Repair, 1.33 mm FOD

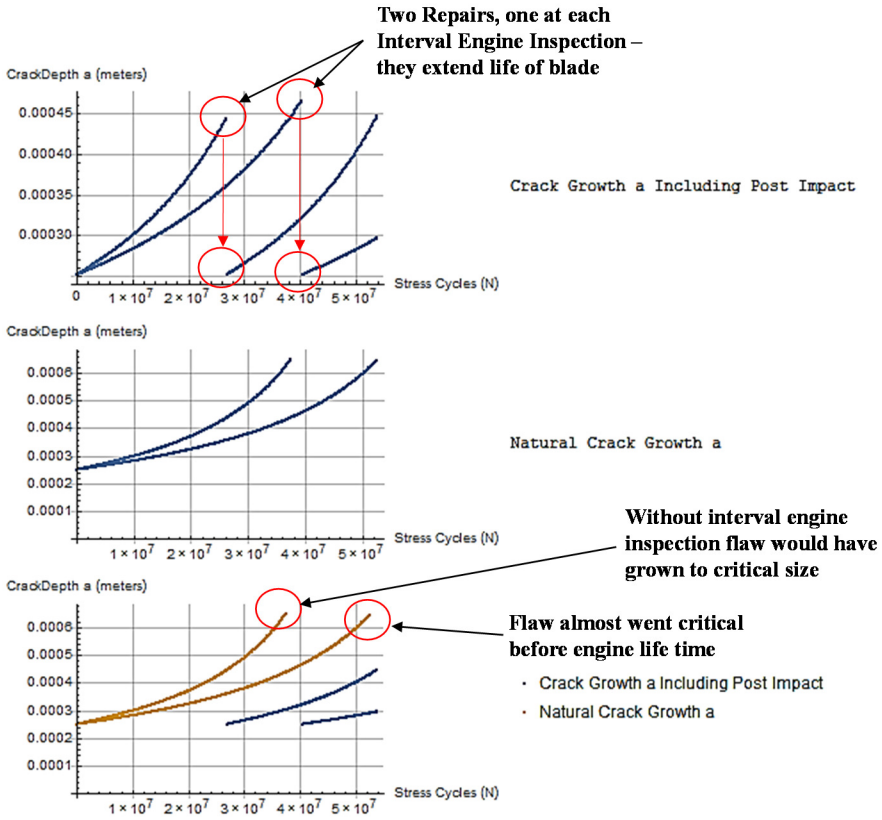


Figure 5.23: DTA Inspect. On, LSP Off, Multi Impact, Repairs, 1.33 mm FOD

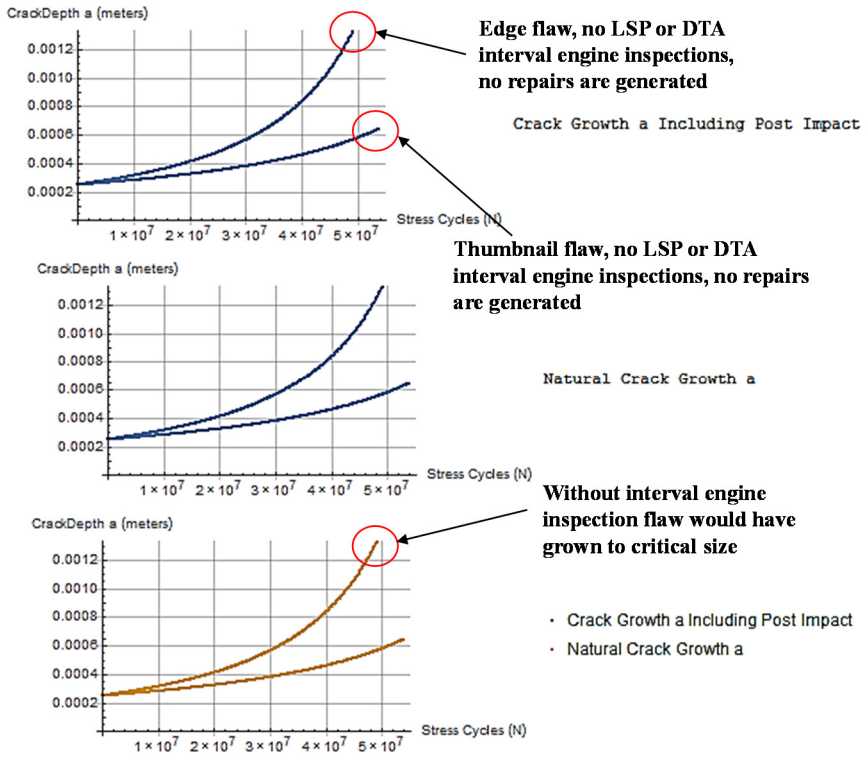


Figure 5.24: DTA Inspect. Off, LSP Off, Multi Impact, No Repairs, 1.33 mm FOD

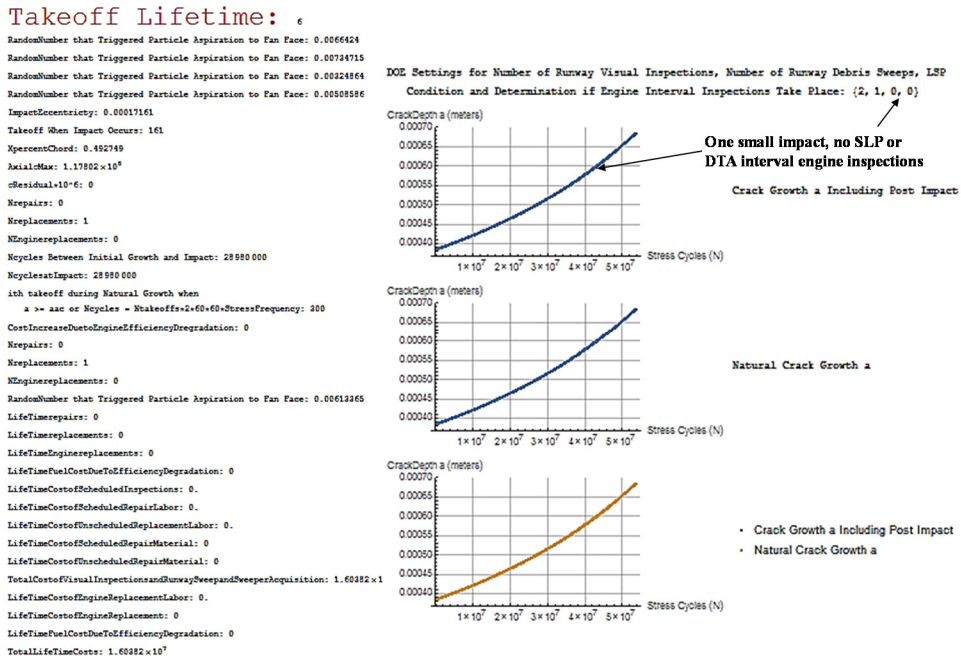


Figure 5.25: DTA Inspect. & LSP Off, Single Impact, No Repair/Repl., 3.2 mm FOD

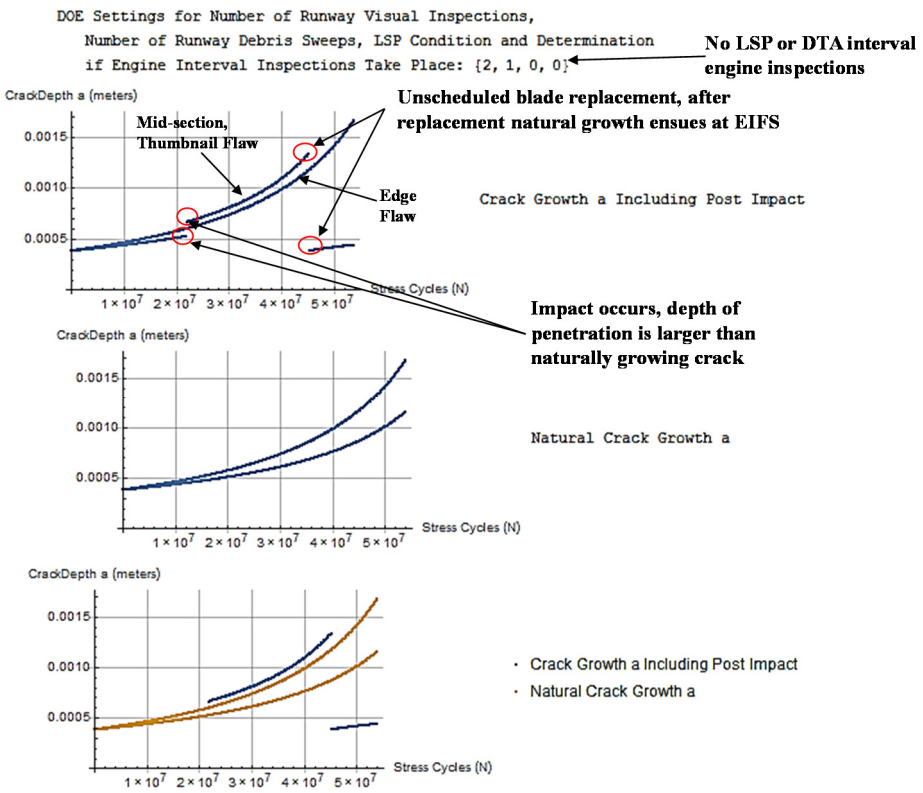


Figure 5.26: DTA Inspect. & LSP Off, Multi Impact, Replacement, 3.2 mm FOD

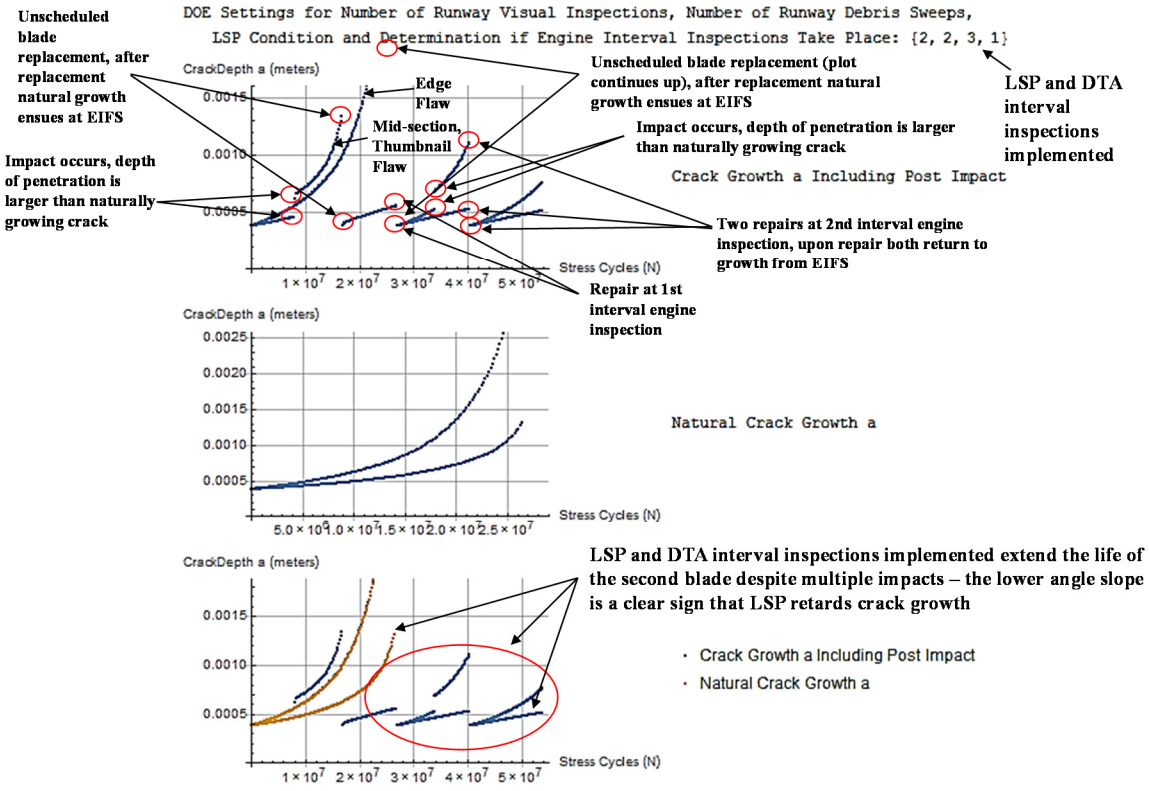


Figure 5.27: DTA Inspect. & LSP On, Full Capability of Simulation, 3.2 mm FOD

LCC simulation does not carry the poignant visual message – instead the ensuing section on dominant experimental factors will shed some light into how expensive and complex keeping engines running safely can be.

5.3.3 DOMINANT CONTRIBUTORS

This section will begin by gaging the importance of each of the factors for both particle sizes under both regimes of inspection on the runway – recall that for visual runway inspections the factors are particle diameter, runway sweeps, number of runway visual inspections, LSP implementation and DTA interval engine inspections while for automated inspections they are particle diameter, runway sweeps, scan rate, probability of automated detection, LSP implementation and DTA interval engine inspections (see Table 5.4 for factor ranges). From 19,392 runs (2 particle sizes, 24 factor settings for visual inspections and 72 settings for automated inspections with each simulation run replicated 100 times) 19,392 outputs each for LRU replacements and repairs and LCCs where recorded. This massive amount of data cannot be reasonably attached in visual format to this thesis, instead the files will be inserted to this document and the screening test results will be presented – the screening tests were performed with the statistical software JMP®, their plots reside in Appendix I (employ Figure 5-16 as rubric to discern statistically significant factors and any higher order terms and interactions). Figures 5-28, 5-29 and 5-30 present the LCC, LRU replacement and repair trends respectively. The salient findings are:

- For the screening test plots in Appendix I:
    - Figure I.1 reveals that the statistically significant factors for LCCs when visual inspections of the runway are performed are particle diameter, Damage Tolerance interval engine inspections and their interaction, the

interaction of Damage Tolerance interval engine inspections and LSP treatment and the interaction of particle diameter, Damage Tolerance interval engine inspections and LSP treatment - LSP, the number of runway visual inspections and sweep are statistically insignificant on their own

- For LCCs when automated runway inspections are considered Figure I.4 reveals that the statistically significant factors are particle diameter, Damage Tolerance interval engine inspections, LSP treatment, scan rate and interactions of these - runway sweeping and the probability of detection from the automated system on their own are largely insignificant
- Figure I.2 reveals that the statistically significant factors for LRU replacements when visual and automated inspections of the runway are performed are particle diameter, Damage Tolerance interval engine inspections, LSP treatment and the interaction of particle diameter and Damage Tolerance interval engine inspections - the number of runway visual inspections, runway sweeps and the scan rate and probability of detection of the automated runway inspection system are statistically insignificant
- Figure I.3 reveals that the statistically significant factors for LRU repairs when visual inspections of the runway are performed are particle diameter, Damage Tolerance interval engine inspections and their interactions - number of visual inspections and sweeps of the runway and LSP are statistically insignificant

- Figure I.6 in reveals that the statistically significant factors for LRU repairs when automated inspections of the runway are performed are particle diameter, Damage Tolerance interval engine inspections, scan rate of the automated runway inspection system and their interactions - number of runway sweeps, the probability of detection and LSP are statistically insignificant

- Figures 5-28, 5-29 and 5-30 reveal that FOD elimination, defined in this thesis as minimizing the FOD-exacerbated HCF failure mode that leads to Primary System replacements and/or catastrophic engine losses while reducing or marginally increasing LCCs, is achieved for the larger particle - though LCCs increase 30% vs. the paradigm, LRU replacements are reduced by 92% vs. the paradigm for the full configuration - LRU repairs are also reduced, but at a slower rate than replacements – these trends address and support **Research Questions 1& 2** and **Hypotheses 1 & 2** respectively in this thesis: elimination of the FOD-exacerbated HCF failure mode is achieved affordably in the life cycle of an engine by deploying/employing automated runway inspections under a nearly continuous scan rate, Laser Shock Peening treatment of the rotor blade surface and Damage Tolerance interval engine inspections for the larger particle considered in this thesis
- Visual inspections and sweep of the runway are ineffective at eliminating the failure mode
- An automated system that is scanning nearly constantly, LSP treatments of the rotor blades in the suitable locations (refer to Section 4.9.2 for LSP treatment

mappings developed from impact data) and Damage Tolerance interval engine inspections provide the most effective, affordable configuration that achieves FOD elimination for the larger particle considered

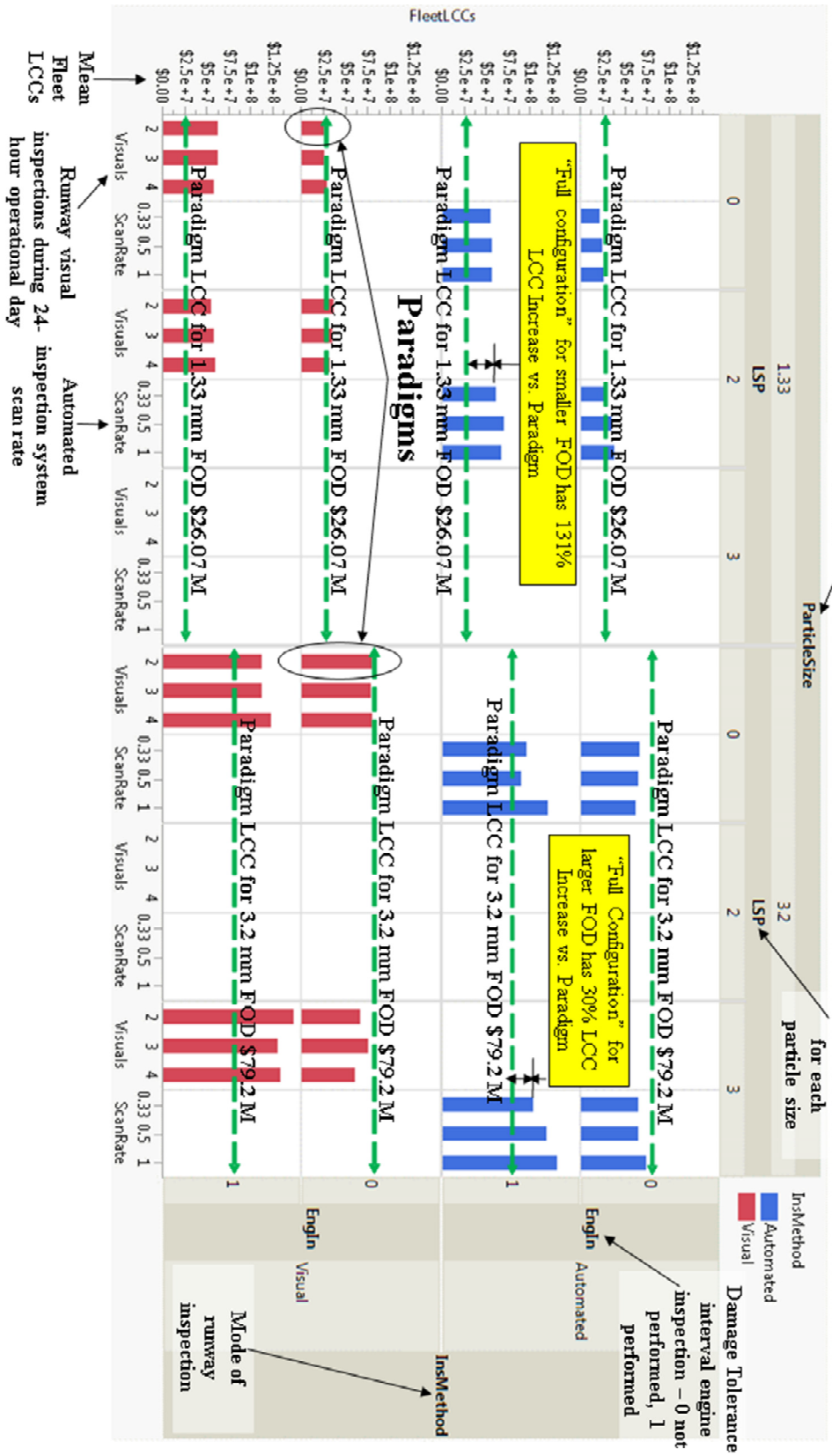


Figure 5.28: Mean of Fleet LCCs vs. Factors

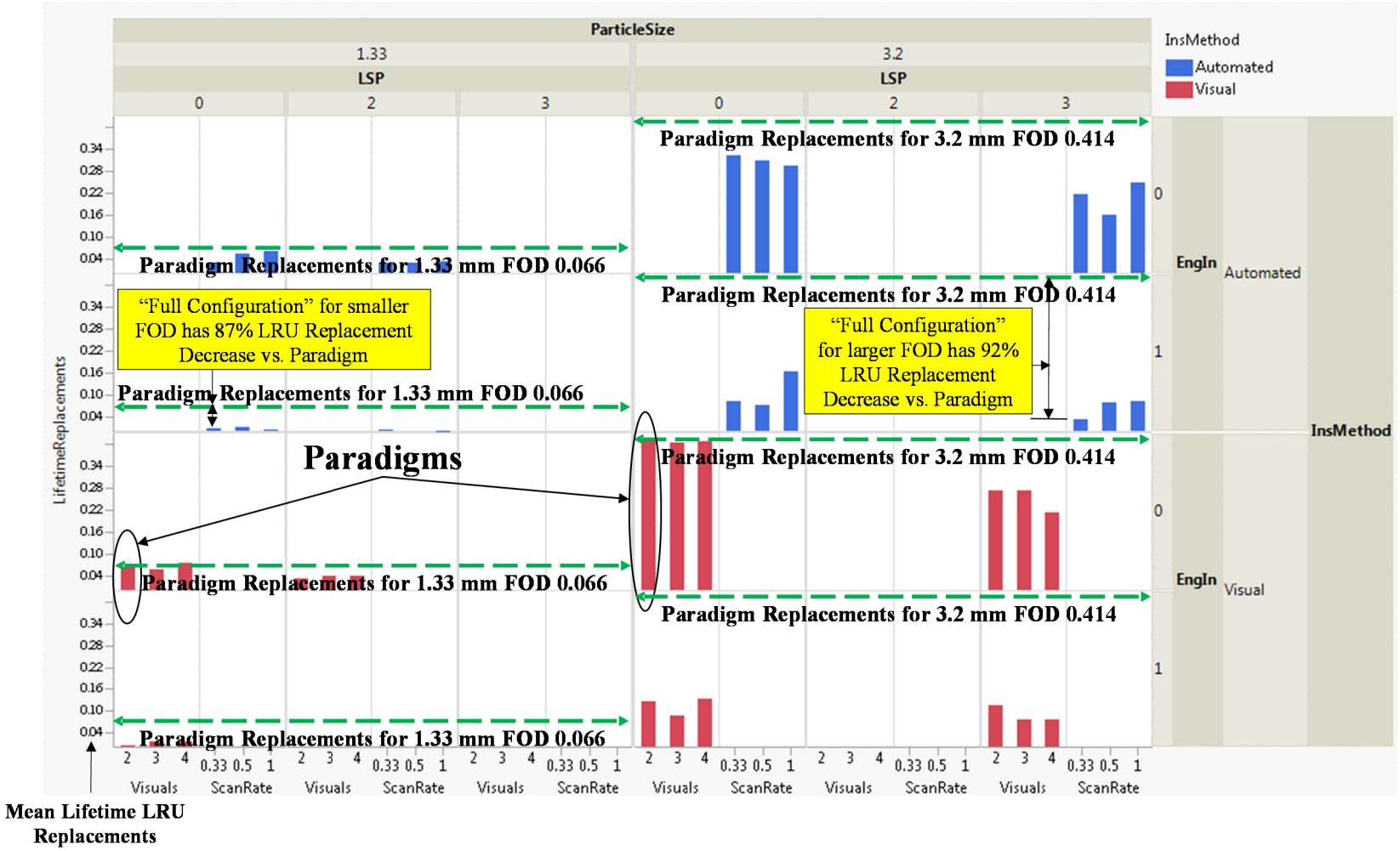


Figure 5.29: Mean of Replacements per Engine vs. Factors

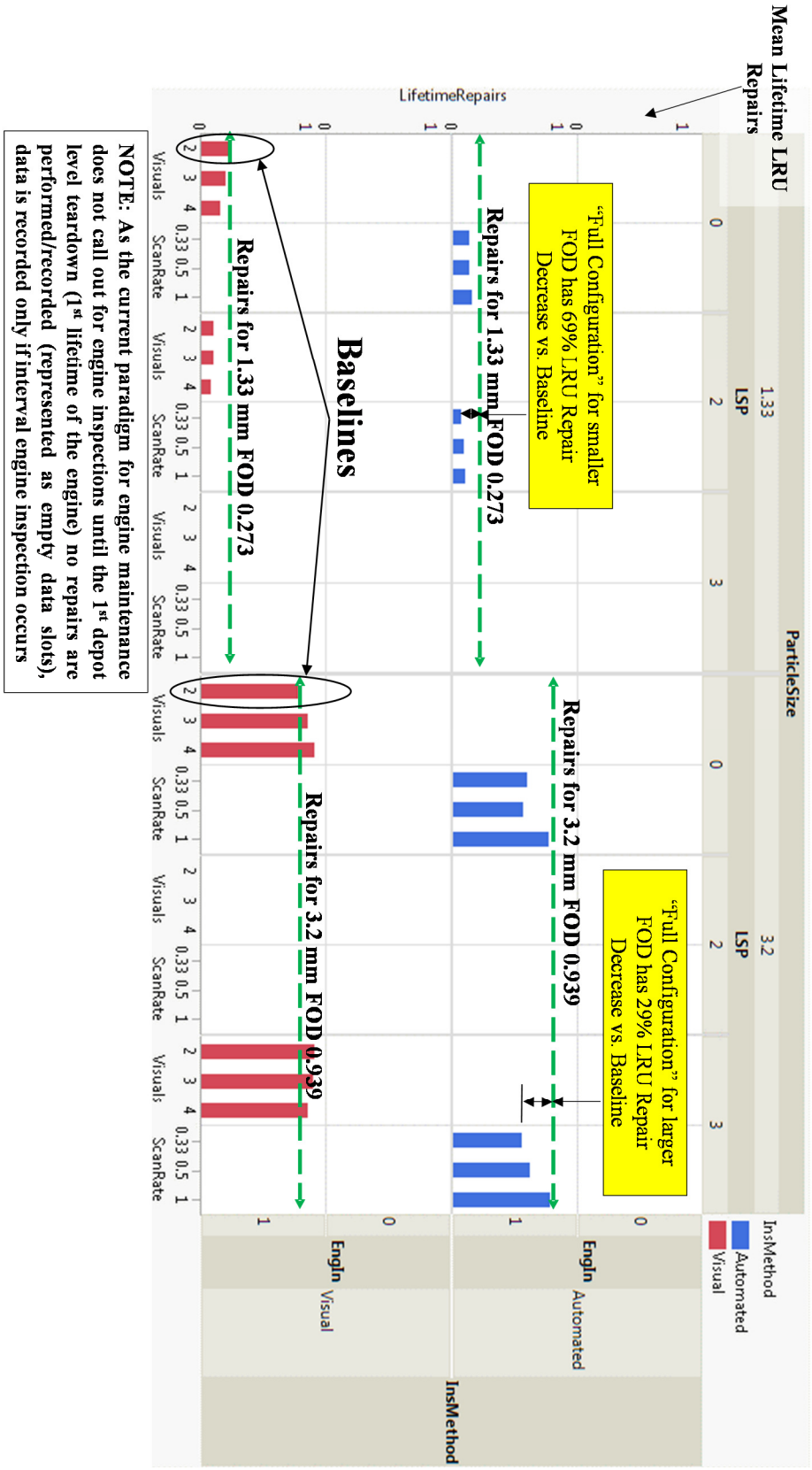


Figure 5.30: Mean of Repairs per Engine vs. Factors

6 CONCLUSIONS & RECOMMENDATIONS

6.1 CONCLUSIONS

Recall that the research objective was to determine if FOD elimination, the minimization of the FOD-exacerbated HCF failure mode that leads to 1st HPC rotor blade replacements and/or catastrophic engine losses while reducing or marginally increasing LCCs, is achievable and if so by what mix of technologies, processes and/or design changes at the primary and support systems. The case study experiments of Section 5 revealed the following conclusions that align to the research objective and address and support the Research Questions and Hypothesis:

- FOD elimination, defined in this thesis as minimizing the FOD-exacerbated HCF failure mode that leads to Primary System replacements and/or catastrophic engine losses while reducing or marginally increasing LCCs, is achieved for the larger particle - though LCCs increase 30% vs. the current paradigm, LRU replacements are reduced by 92% vs. the paradigm for the full configuration - LRU repairs are also reduced, but at a slower rate than replacements – these trends address and support **Research Questions 1** and **Hypotheses 1** respectively in this thesis: elimination of the FOD-exacerbated HCF failure mode is achieved affordably in the life cycle of an engine by deploying/employing automated runway inspections under a nearly continuous scan rate, Laser Shock Peening treatment of the rotor blade surface and Damage Tolerance interval engine inspections for the larger particle considered in this thesis

- Extending radially outward from 60% of the rotor span, on the Suction Side of the airfoil larger particle impacts occur more often near the Leading Edge of the airfoil than impacts of the smaller particle (non-dimensional chord locations that range from 0 to 0.95 for the smaller particle vs. 0 to 0.16 for the larger one) - with the same radial span on the Pressure Side of the airfoil impacts occur across the chord for both particle sizes - these trends address and support Research Question 2 and Hypothesis 2 respectively: with the right mix and placement of countermeasures (automated runway debris detection system at near continuous scan rate, implementation of LSP at the identified locations on the airfoil and interval engine inspections) the methodology presented in this thesis estimate LCCs that are marginally higher (30%) and LRU replacement rates that are lower (92% lower) vs. LCCs and rate of LRU replacement of the current paradigm for the larger particle
- Upon impact the smaller particle causes smaller penetration depths – this along with the discovery that the mode of impact is different if it occurs on the Pressure or Suction side of the airfoil (on the Pressure Side of the airfoil the impact velocity is more severe because the event is similar to a "paddle" strike with a very fast rotating rotor striking the particle, whereas on the Suction Side it is the slower velocity at impact due to the particle striking the blade that occurs) addresses and supports Research Question 3 and Hypothesis 3 respectively in this thesis – indeed there are factors that influence LCC affordability more significantly vs. other factors that were also varied during the experiments: the smaller particle impacts the rotor blades more often, but

their depths of penetration are smaller, imparting stress intensities (via Beta geometry factors that are more benign) at the point of impact that will grow cracks more slowly, causing less damage that leads to fewer blade replacements and reduced LCCs

- For the two particles sizes considered in the case study of this thesis the assertion of dominant leading edge impacts assumed for coupon testing in the public domain literature is debunked - recall that most tests that simulate particle impact on axial compressor blades are performed on the leading edges of specimens - the mappings of impacts developed for this research effort paint a different story, impact is the stuff of distributions for rotating blades
- The longer the ground distance from the stagnation point to the fan face the more time the particle has to convert more of its velocity to an axial component and because the flow meridional velocity does not change appreciably across the engine stations higher particle velocities inside the engine occur, this relationship along with a higher rate of impact at the Fan station by the larger particle corroborates that the smaller particle that moves more quickly inside the engine will impact the 1st HPC rotor blades more often – the smaller, faster moving particle more often misses the slower rotating fan (Ω of 10,000 rpm at fan vs 14,000 rpm at HPC rotor) and reaches the faster rotating HPC rotor where it has a greater rate of impact and of passing through that station with no impact than the larger particle - this finding hints at a System of Systems solution, perhaps if the aircraft performed a rolling takeoff from the tarmac as it reached the area of greatest risk, the runway, it achieved enough effective Head Wind

to move the stagnation point closer to the fan face to reduce the particle velocity at inlet or to entirely blow away the vortex that in the first place caused ground aspiration

- Size truly matters - the larger particle strikes the rotor blades less often, but the ensuing damage and LCC/LRU replacement implications are greater vs. the smaller particle - it is precisely for the larger particle that the full configuration of technology/process/design changes has the greatest impact to affordability
- Probability of Detection for the automated runway detection system considered (0.65 to 0.74) was statistically insignificant toward LCCs, LRU replacements and repairs - for an automated system it is continuous or near continuous scan rate that drives their effectivity for dispatching FOD removal from a runway, i.e. scanning quickly and often
- Visual inspections and runway sweeps for the two particle sizes considered proved ineffective at eliminating the FOD-exacerbated HCF failure mode - they effectively remove very little FOD from a runway during a 24-hour operational day - automated inspections along with LSP and DTA interval engine inspections reduce significantly the risk of engine catastrophic damage because flaws that would have gone critical are detected and repaired, thus allowing a full realization of the blade/engine design life

6.2 RECOMMENDATIONS

The following are the salient recommendations that resulted from this research effort:

- At a strategic level: infuse knowledge early into the conceptual design of engines/engines on aircraft – by employing the methodology and tools developed in this thesis disciplines such as Fracture Mechanics can be brought forth to shed light early in the design process by leveraging what is known/documented (presence, distribution and nature of FOD on the runway along with ambient/operational conditions and likely engine geometry, as military low bypass turbofans do not change drastically in design) to estimate the LCC and LRU replacement implications of changes/enhancements at the Primary and Support Systems
- At a tactical level:
    - Invest in LSP – it's LCC impact is small, but its benefit to reducing Replacements is large
    - Performing DTA interval engine inspections – it impacts LCCs significantly, but the impact to reducing LRU replacements is larger
    - Go for Automation – automated inspections have LCCs comparable to Visual inspections but are more accurate and precise thus having a greater impact on reducing LRU replacements
    - Leverage impact mappings – employing impact mappings will determine where surface treatments such as LSP are needed most on the airfoil and where no treatment is necessary
- If OEMs and operators decided to not pursue FOD elimination in a systematic, aggressive manner such as is set forth in this thesis, the following consequences will continue to plague the industry:

- Continued design of engines/engines on aircraft that are not FOD robust and that will increasingly operate in FOD prone areas around the globe – recall that Russian aircraft in the Syrian theater of operation had a 40% drop in readiness due to FOD from desert conditions
- Continued operation of aircraft/engines under a paradigm that is inefficient and not affordable
- The consequences of FOD are not the cost of doing business as was said by a top executive at an OEM – 1st level approximations to the affect/effect of FOD can be made by implementing the methodology set forth in this thesis that will lead to more affordable, robust engine/engine on aircraft designs
- Additionally great LLC savings could be missed by not pursuing FOD elimination systematically as it is showcased in this thesis – the cost of not being able to perform missions, of increased spares and unscheduled blade replacements and of perceived un-robust engine/engine on aircraft designs are likely to be large

APPENDIX A: VORTEX PRESSURE RATIOS & RAW DATA

Raw data is presented in this appendix on engine height to inlet diameter (h/D_l), inlet Mach numbers, velocity ratio ($U_i/U_{\infty,}$ inlet to headwind velocity), ground roll velocity (U_G), non-dimensional vortex strength (Γ^*, average vortex strength $\bar{\Gamma}$ divided by product of inlet velocity U_i and inlet diameter D_l) and ambient laboratory conditions from Murphy's work on the aerodynamics of ground vortices [96]. The magnitude of pressure ratios from Murphy's scaled engine inlet are kept for the full scale inlet considered in this thesis – the assumption is supported by dynamic similarity. The raw data enables the calculation of stagnation point and fan face pressure ratios via the methodology presented in Section 4.4.

Table A.1: Murphy's Wind Tunnel Conditions

Murphy's Wind Tunnel Conditions [96]													
T_{amb} (°K)	290	T_i (K)	271.65	P_{amb} (Pa)	96952	ρ_{amb} (kg/m^3)	1.165	D_l (m)	0.1	r_c (vortex core radius, m)	0.0025	Murphy's Model Scale	30

Table A.2: Quiescent Pressure Ratios and Raw Data

Quiescent Conditions												
h/D_l	M_i	U_i (m/s)	U*	U_∞ (m/s)	M_∞	U_{UG} (m/s)	M_{UG}	Γ*	Γ (m^2/s)	ω (rads/s)	P/P∞ (Stag. Point)	P/P∞ (Fan Face)
0.25	0.58	192.10	∞	0	0	0	0	0.07	1.3447	34243	0.934	0.985
0.32	0.58	192.10	∞	0	0	0	0	0.05	0.9605	24459	0.966	
0.4	0.58	192.10	∞	0	0	0	0	0.039	0.755	19225	0.979	0.985
0.25	0.43	142.06	∞	0	0	0	0	0.068	0.9626	24513	0.966	0.995
0.35	0.43	142.06	∞	0	0	0	0	0.044	0.6251	15917	0.986	
0.4	0.43	142.06	∞	0	0	0	0	0.041	0.5825	14832	0.988	0.992
0.29	0.14	46.25	∞	0	0	0	0	0.062	0.2857	7274	0.997	0.997
0.5	0.14	46.25	∞	0	0	0	0					0.997

Table A.3: Headwind Pressure Ratios & Raw Data

Headwind Conditions													
h/D_l	δ^*/D_l (Non-Dim Bndry. Layer Thick.)	M_i	U_i (m/s)	U*	U_∞ (m/s)	M_∞	U_{UG} (m/s)	M_{UG}	Γ*	Γ	ω (rads/s)	P/P∞ (Stag. Point)	P/P∞ (Fan Face)
0.25	0.11	0.58	189.76	19.81	9.58	0.03	0	0	0.066	1.25	31893	0.943	0.960
0.25	0.11	0.58	191.84	13.29	14.434	0.04	0	0	0.077	1.48	37616	0.920	
0.25	0.11	0.58	191.19	10.02	19.085	0.06	0	0	0.111	2.12	54043	0.836	0.960
0.25	0.07	0.58	192.22	19.95	9.633	0.03	0	0	0.06	1.15	29369	0.951	
0.25	0.07	0.58	191.53	13.48	14.213	0.04	0	0	0.083	1.59	40482	0.908	
0.25	0.07	0.58	191.78	10.01	19.168	0.06	0	0	0.144	2.76	70323	0.721	
0.25	0.03	0.58	193.29	20.66	9.357	0.03	0	0	0.033	0.64	16243	0.985	
0.25	0.03	0.58	193.29	13.79	14.015	0.04	0	0	0.067	1.30	32979	0.939	
0.25	0.03	0.58	191.18	10.16	18.822	0.06	0	0	0.097	1.85	47222	0.874	
0.32	0.11	0.58	192.22	19.95	9.633	0.03	0	0	0.059	1.13	28879	0.953	
0.32	0.11	0.58	191.78	10.01	19.168	0.06	0	0	0.136	2.61	66416	0.752	
0.4	0.11	0.58	192.24	19.12	10.057	0.03	0	0	0.062	1.19	30351	0.948	
0.4	0.11	0.58	191.84	13.29	14.434	0.04	0	0	0.113	2.17	55203	0.828	
0.4	0.11	0.58	192.06	9.89	19.414	0.06	0	0	0.167	3.21	81677	0.624	
0.4	0.07	0.58	192.22	19.95	9.633	0.03	0	0	0.069	1.33	33774	0.936	
0.4	0.07	0.58	191.53	13.48	14.213	0.04	0	0	0.117	2.24	57065	0.817	
0.4	0.07	0.58	191.78	10.01	19.168	0.06	0	0	0.157	3.01	76671	0.669	
0.4	0.03	0.58	192.40	20.49	9.392	0.03	0	0	0.063	1.21	30867	0.946	
0.4	0.03	0.58	190.00	13.56	14.015	0.04	0	0	0.156	2.96	75478	0.679	
0.25	0.11	0.43	135.00	11.58	11.655	0.03	0	0	0.101	1.36	34721	0.932	
0.25	0.11	0.43	135.00	8.36	16.145	0.05	0	0	0.172	2.32	59131	0.803	0.980
0.4	0.11	0.43	135.00	11.56	11.681	0.03	0	0	0.172	2.32	59128	0.803	
0.4	0.11	0.43	134.99	8.30	16.256	0.05	0	0	0.172	2.32	59125	0.803	
2.63	N/A	0.483115	164.91	26.35	6.25845	0.02	0	0	0.0392	1.90	48461	0.866	N/A

Note: Cell highlighted in yellow contains data from NACA TN 3330 - it presents a model verification point for stagnation pressure ratio from a source other than Murphy [96] – the pressure ratio is comparable to those calculated with this thesis' methodology.

Table A.4: Headwind/Ground Roll (U_G) Pressure Ratios & Raw Data

Headwind and Ground Speed												
h/D_l	M_i	U_i	U*	U∞	M∞	U_{UG}	M_{UG}	Γ*	Γ	ω (rads/s)	P/P∞ (Stag. Point)	P/P∞ (Fan Face)
0.25	0.58	193.3937	19.52	9.91	0.03	10.00	0.03	0.06	1.14	29056	0.952	0.960
0.25	0.58	192.7842	12.68	15.21	0.04	15.00	0.04	0.08	1.47	37310	0.922	0.970
0.25	0.58	192.7822	9.50	20.30	0.06	20.00	0.06	0.08	1.58	40255	0.909	0.980
0.25	0.58	189.7606	19.81	9.58	0.03	0.00	0.00	0.07	1.27	32376	0.941	0.950
0.25	0.58	192.1	∞	0.00	0.00	0.00	0.00	0.06	1.13	28862	0.953	0.960
0.25	0.58	193.2387	12.40	15.58	0.05	5.00	0.01	0.08	1.55	39366	0.913	
0.25	0.58	192.9736	9.27	20.82	0.06	10.00	0.03	0.09	1.64	41769	0.902	
0.25	0.58	192.8365	10.02	19.25	0.06	0.00	0.00	0.11	2.18	55489	0.827	
0.4	0.58	187.1804	19.03	9.83	0.03	10.00	0.03	0.07	1.31	33366	0.937	
0.4	0.58	191.5538	12.68	15.10	0.04	15.00	0.04	0.12	2.38	60486	0.794	

APPENDIX B: VORTEX AT INLET LOCATIONS

Data is presented in Appendix A for the pressure ratios of the ingested vortices modeled by Murphy, the associated vertical locations on the inlet are presented in this appendix [96]. The vertical location of the vortices at the inlet are scaled/corrected per Equation 23 presented in Section 4.4. Equation B.1 is the fit of the corrected, full scale height of the vortex at the inlet and Figure B.1 its Goodness of Fit measures.

Table B.1: Quiescent Full Scale, Corrected Location of Vortex at Inlet

Quiescent Conditions									
h/D_l	δ/D_l (Non-Dim Boundary Layer Thickness)	M_i	U_i	U*	$U\infty$	U_{UG}	Γ*	Non-Dim Height of Vortex at Fan Face (Corrected Z/Dl)	Full Scale Height of Vortex at Fan Face (m)
0.25	0	0.58	192.10	∞	0.00	0.00	0.07	0.31	0.27
0.4	0	0.58	192.10	∞	0.00	0.00	0.04	0.46	0.41
0.25	0	0.43	142.06	∞	0.00	0.00	0.07	0.40	0.36
0.4	0	0.43	142.06	∞	0.00	0.00	0.04	0.58	0.51

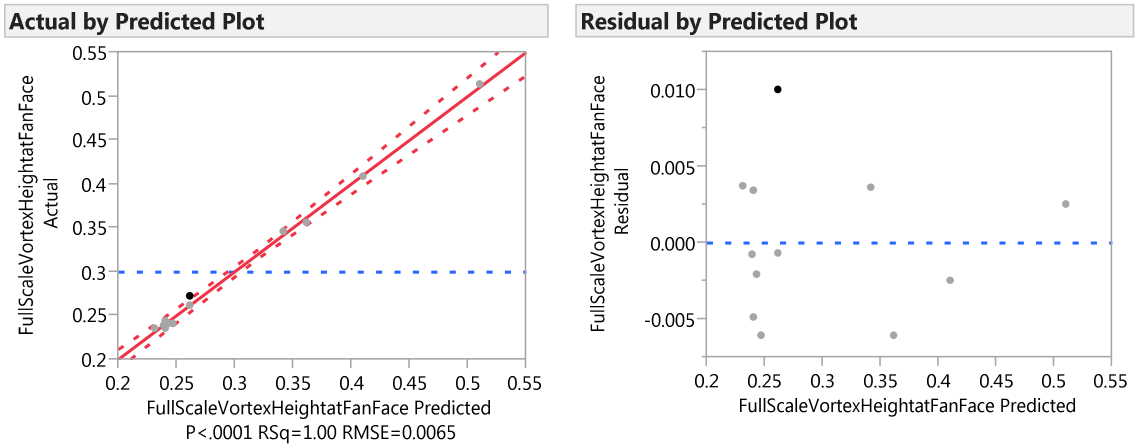
Table B.2: Headwind Full Scale, Corrected Location of Vortex at Inlet

HeadWind Conditions									
h/D_l	δ/D_l (Non-Dim Boundary Layer Thickness)	M_i	U_i	U*	$U\infty$	U_{UG}	Γ*	Non-Dim Height of Vortex at Fan Face (Corrected Z/Dl)	Full Scale Height of Vortex at Fan Face (m)
0.25	1.03	0.58	191.19	10.02	19.09	0.00	0.11	0.27	0.24
0.25	1.03	0.43	135.00	8.36	16.15	0.00	0.17	0.39	0.35
0.25	1.03	0.58	189.76	19.81	9.58	0.00	0.07	0.27	0.24
0.25	1.03	0.58	191.25	23.91	8.00	0.00	0.19	0.28	0.24

Table B.3: Headwind/Rolling Ground Full Scale, Corrected Vortex at Inlet Location

Headwind/Rolling Ground Conditions									
h/D_l	δ/D_l (Non-Dim Boundary Layer Thickness)	M_i	U_i	U*	$U\infty$	U_{UG}	Γ*	Non-Dim Height of Vortex at Fan Face (Corrected Z/Dl)	Full Scale Height of Vortex at Fan Face (m)
0.25	0	0.58	192.10	∞	0.00	0.00	0.06	0.30	0.26
0.25	0	0.58	193.39	19.52	9.91	10.00	0.06	0.27	0.24
0.25	0	0.58	192.78	12.68	15.21	15.00	0.08	0.27	0.24
0.25	0	0.58	192.78	9.50	20.30	20.00	0.08	0.27	0.24

$$UzofVortexatFanFace\_FullScale = 0.400409467782324 + 0.990523720375341 \cdot h/D_l - 0.221834354850636 \cdot NonDimBoundaryLayerThickness - 0.664790011747447 \cdot InletMachNumber + 0.000266937968596887 \cdot HeadWind - 0.00179140669537652 \cdot URollingGround \quad (B.1)$$



Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	0.4004095	0.022355	17.91	<.0001*
h/Dl	0.9905237	0.039613	25.01	<.0001*
δ/Dl	-0.221834	0.099392	-2.23	0.0671
Mi	-0.66479	0.032102	-20.71	<.0001*
Vwind	0.0002669	0.000734	0.36	0.7284
UG	-0.001791	0.000838	-2.14	0.0765

Summary of Fit

RSquare	0.997064
RSquare Adj	0.994618
Root Mean Square Error	0.006518
Mean of Response	0.299488
Observations (or Sum Wgts)	12

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
h/Dl	0.9905237	0.039613	25.01	<.0001*
Mi	-0.66479	0.032102	-20.71	<.0001*
δ/Dl	-0.221834	0.099392	-2.23	0.0671
UG	-0.001791	0.000838	-2.14	0.0765
Vwind	0.0002669	0.000734	0.36	0.7284

Figure B.1: Goodness of Fit Measures for Corrected Vortex Location at Inlet

APPENDIX C: ASPIRATION SIMULATION CODE

The code presented in this appendix was developed in Wolfram Mathematica® 10. The code applies to the case of a steel, spherical FOD particle 1.33 mm in diameter aspirated from the ground to fan face under conditions of Headwind (U_∞) velocity (with range of 9.357 m/s to 12 m/s) equaling Rolling Ground (UG) for takeoff velocity – this combination of equal Headwind and Rolling Ground rid the possibility of a boundary layer. The code can run other particle diameters by simply updating the d_{fod} variable for other particle diameters. The code relies on the inlet static temperature distribution that was developed with the Meanline Analysis tool of Section 4.6. The code by means of input parameter settings (engine ground clearance, horizontal distance to fan face, inlet Mach number and Headwind) arranged/set from a full factorial and Latin Hypercube space filling Design of Experiments (DOE) extracts inlet velocities (leveraged as Response Surface fits) that are the inlet initial conditions for velocity in the In-Engine Particle kinetics model of Section 4.7. The vortex ingestion/particle aspiration vertical location at inlet/fan face developed from scaled/corrected data from Murphy's work [96] along with setting the displacement at start to zero and the horizontal distance from stagnation point to fan face set via DOE setting provide the initial conditions.

Aspiration Simulation Code – 1.33 mm Steel Debris

```
ClearAll["Global`*"]
Off[Power::infy]
ClearAll[x, y, Minlet, Vwind, Ug, HoDinlet, InletMachNumber,
 HeadWind, URollingGround, HoDl, t, Tf, a, b, Tfx, Tfy, ax, ay, bx, by,
 XX1, YY1, sol2, tx1, tx, InletMachNumberAppended, HeadWindAppended,
 URollingGroundAppended, PRvortexongroundAppended, NonDimVortexStrengthAppended,
 PRvortexonfanfaceAppended, UxFanFace, UyFanFace, VxFanFace, VyFanFace, Counter]
"Statistics`NonlinearFit`"
AppendTo[$Path, ToFileName[
   {$HomeDirectory, "GeorgiaTech\\Research\\MyResearch\\MathematicaRuns"}]];
SteelParticleInput = Import["FOD Inlet Velocity
    Model_Steel133mm_4Factors_NewModelUy_NoBoundaryLayer_Input.csv", "CSV"]
with[{g = 9.8, Cdsphere = 0.47, Pambsealevel = 101 325, Rhosealevel = 1.225}];
(* Murphy's Data to Model Vortex Strength *)
data1 = {{0.25, 0.581452381538234, 0, 0, 0.07},
   {0.32, 0.581452381538234, 0, 0, 0.05}, {0.4, 0.581452381538234, 0, 0, 0.0393},
   {0.25, 0.43, 0, 0, 0.06776}, {0.35, 0.43, 0, 0, 0.044}, {0.4, 0.43, 0, 0, 0.041},
   {0.29, 0.14, 0, 0, 0.06176}, {0.25, 0.581452381538234, 9.58, 0, 0.066},
   {0.25, 0.581452381538234, 14.434, 0, 0.077}, {0.25, 0.581452381538234,
    19.085, 0, 0.111}, {0.25, 0.581452381538234, 9.633, 0, 0.06},
   {0.25, 0.581452381538234, 14.213, 0, 0.083}, {0.25, 0.581452381538234,
    19.168, 0, 0.144}, {0.25, 0.581452381538234, 9.357, 0, 0.033},
   {0.25, 0.581452381538234, 14.015, 0, 0.067}, {0.25, 0.581452381538234,
    18.822, 0, 0.097}, {0.32, 0.581452381538234, 9.633, 0, 0.059},
   {0.32, 0.581452381538234, 19.168, 0, 0.136}, {0.4, 0.581452381538234,
    10.057, 0, 0.062}, {0.4, 0.581452381538234, 14.434, 0, 0.113},
   {0.4, 0.581452381538234, 19.414, 0, 0.167}, {0.4, 0.581452381538234,
    9.633, 0, 0.069}, {0.4, 0.581452381538234, 14.213, 0, 0.117},
   {0.4, 0.581452381538234, 19.168, 0, 0.157}, {0.4, 0.581452381538234,
    9.392, 0, 0.063}, {0.4, 0.581452381538234, 14.015, 0, 0.156},
   {0.4, 0.581452381538234, 18.819, 0, 0.185}, {0.25, 0.43, 11.655, 0, 0.101},
   {0.25, 0.43, 16.145, 0, 0.172}, {0.4, 0.43, 11.681, 0, 0.172},
   {0.4, 0.43, 16.256, 0, 0.172}, {0.25, 0.581452381538234, 9.91, 10, 0.059},
   {0.25, 0.581452381538234, 15.205, 15, 0.076},
   {0.25, 0.581452381538234, 20.295, 20, 0.082},
   {0.25, 0.581452381538234, 9.58, 0, 0.067}, {0.25, 0.581452381538234,
    15.58, 5, 0.08}, {0.25, 0.581452381538234, 20.817, 10, 0.085},
   {0.25, 0.581452381538234, 19.249, 0, 0.113}, {0.4, 0.581452381538234,
    9.834, 10, 0.07}, {0.4, 0.581452381538234, 15.102, 15, 0.124}};
(* Murphy's Data to Model Stagnation Point Pressure Ratio *)
data2 = {{0.25, 0.581452381538234, 0, 0, 0.933962226182042},
   {0.32, 0.581452381538234, 0, 0, 0.966307258256144},
```

```
    {0.4, 0.581452381538234, 0, 0, 0.979184758921613},
    {0.25, 0.43, 0, 0, 0.966158333375301}, {0.35, 0.43, 0, 0, 0.985730449222171},
    {0.4, 0.43, 0, 0, 0.98760996133392}, {0.29, 0.14, 0, 0, 0.997019846163257},
    {0.25, 0.581452381538234, 9.91, 10, 0.952452240407922},
    {0.25, 0.581452381538234, 15.205, 15, 0.921600796241903},
    {0.25, 0.581452381538234, 20.295, 20, 0.908735221075807},
    {0.4, 0.581452381538234, 9.834, 10, 0.937301343836281}};
(* Murphy's Data to Model Fan Face Pressure Ratio *)
data3 =
  {{0.25, 0.581452381538234, 0, 0, 0.985}, {0.4, 0.581452381538234, 0, 0, 0.985},
   {0.25, 0.43, 0, 0, 0.995}, {0.4, 0.43, 0, 0, 0.992}, {0.29, 0.14, 0, 0, 0.997},
   {0.5, 0.14, 0, 0, 0.997}, {0.25, 0.581452381538234, 9.58, 0, 0.96},
   {0.25, 0.581452381538234, 19.085, 0, 0.96}, {0.25, 0.43, 16.145, 0, 0.98},
   {0.25, 0.581452381538234, 9.91, 10, 0.96}, {0.25, 0.581452381538234,
    15.205, 15, 0.97}, {0.25, 0.581452381538234, 20.295, 20, 0.98},
   {0.25, 0.581452381538234, 9.58, 0, 0.95}, {0.25, 0.581452381538234, 0, 0, 0.96},
   {0.25, 0.581452381538234, 19.168, 0, 0.96},
   {0.25, 0.581452381538234, 23.906, 0, 0.95}};
SteelParticleFullFactorialOutput = {};
Do[

 Dinlet = 0.88;
 Dshaft = 0.42;
 InletMachNumberAppended = {};
 HeadWindAppended = {};
 URollingGroundAppended = {};   (*1st Model Append Variables*)
 BoundaryLayerNonDimAppended = {};
 PRvortexongroundAppended = {};
 NonDimVortexStrengthAppended = {};
 PRvortexonfanfaceAppended = {};
 UzofVortexatFanFaceAppended = {};
 tx1 = {};
 ax1 = {};
 bx1 = {};
 UxFanFace = {};
 UyFanFace = {};
 VxFanFace = {};
 VyFanFace = {};
 YatFanFace = {};
 TfxAppended = {};
 DataSetAppended = {};
 StoredMeanVxFanFace = {};
 StoredMeanVzFanFace = {};
```

Aspiration Simulation Code – 1.33 mm Steel Debris (continued)


```
acoeffcientforVyInitial = {};
bcoeffcientforVxInitial = {};
UzStdDeviationofMean = {};
VzStdDeviationofMean = {};
VxStdDeviationofMean = {};
acoeffcientforVyInitialStdDeviationofMean = {};
bcoeffcientforVxInitialStdDeviationofMean = {};

VortexonGroundPRatio = {};
NonDimensionalVortexStrength = {};
VortexonFanFacePRatio = {};
StoredMeanVortexonGroundPRatio = {};
StoredMeanNonDimensionalVortexStrength = {};
StoredMeanVortexonFanFacePRatio = {};
VortexonGroundPRatioStdDeviationofMean = {};
NonDimensionalVortexStrengthStdDeviationofMean = {};
VortexonFanFacePRatioStdDeviationofMean = {};
StoredMeanacoeffcientforVyInitial = {};
StoredMeanbcoeffcientforVxInitial = {};

kCounter = {}; (*Counter of Sample Size*)
StartingSampleSize = 1;
MaxSampleSize = Length[SteelParticleInput];
SampleSizeIncrement = 1;

(* How Mathematica does nonlinear fits *)
soln1 = NonlinearModelFit[data1,
  II * HoDinlet * Minlet + J * HoDinlet * Vwind + KK * HoDinlet * Ug +
   L * Minlet * Vwind + M * Minlet * Ug + NN * Vwind * Ug + A * HoDinlet^2 + B * HoDinlet +
   CC * Minlet^2 + DD * Minlet + EE * Vwind^2 + F * Vwind + G * Ug^2 + GG * Ug + H,
  {A, B, BB, CC, DD, EE, F, G, GG, H, II, J, KK, L, M, NN},
  {HoDinlet, Minlet, Vwind, Ug}];
soln2 = NonlinearModelFit[data2, AA * HoDinlet * Minlet +
   CC * HoDinlet * Ug + EE * Minlet * Vwind + FF * Minlet * Ug +
   LL * HoDinlet + MM * Minlet^2 + NN * Minlet + PP * Vwind + RR * Ug + UU,
  {AA, CC, EE, FF, LL, MM, NN, PP, RR, UU}, {HoDinlet, Minlet, Vwind, Ug}];
soln3 = NonlinearModelFit[data3, J * HoDinlet * Vwind + KK * HoDinlet * Ug + H,
  {H, J, KK}, {HoDinlet, Minlet, Vwind, Ug}];

Do[m = k; n = k;
 Clear[Tfx, ax, bx, tx, Tf, a, b, t];
 InletMachNumber = SteelParticleInput[[k, 3]](*0.45+RandomNumber*0.05*);
 (*NEEDS TO COME FROM MEANLINE*)
```

```
RandomNumber = (InletMachNumber - 0.45) / 0.05;
ϱMaterial = 7850;
FodGroundDisttoFan = SteelParticleInput[[k, 2]];
dfod = 0.00133;
rfod = dfod / 2;
A0 = Pi * rfod^2;
mimpactor = ϱMaterial * 4 / 3 * Pi * rfod^3;
h = SteelParticleInput[[k, 1]];
HoDl = h / Dinlet;
(*Inlet Static Temperature model comes from MEANLINE model*)
AlphaInletStaticTemp = 0.963476387554866;
BetaInletStaticTemp = 1.01802063;
MaxInletStaticTemp = 306.2104127;
MinInletStaticTemp = 298.3426659;
InletStaticTemp = InverseBetaRegularized[
     RandomNumber, AlphaInletStaticTemp, BetaInletStaticTemp] *
    (MaxInletStaticTemp - MinInletStaticTemp) + MinInletStaticTemp;
InletMachNumberAppended = AppendTo[InletMachNumberAppended, InletMachNumber];
HeadWind = SteelParticleInput[[k, 4]];
HeadWindAppended = AppendTo[HeadWindAppended, HeadWind];
URollingGround = If[HeadWind ≥ 9, HeadWind, 0];
NonDimBoundaryLayerThickness = 0;
URollingGroundAppended = AppendTo[URollingGroundAppended, URollingGround];
NonDimVortexStrength = soln1[HoDl, InletMachNumber, HeadWind, URollingGround];
NonDimVortexStrengthAppended =
 AppendTo[NonDimVortexStrengthAppended, NonDimVortexStrength];
PRvortexonground = soln2[HoDl, InletMachNumber, HeadWind, URollingGround];
PRvortexongroundAppended = AppendTo[PRvortexongroundAppended, PRvortexonground];
(*PRvortexonfanface=If[soln3[HoDl,InletMachNumber,HeadWind,URollingGround]<=1,
   soln3[HoDl,InletMachNumber,HeadWind,URollingGround],1];*)
PRvortexonfanface = soln3[HoDl, InletMachNumber, HeadWind, URollingGround];
UzofVortexatFanFace =
 0.400409467782324 + 0.990523720375341 * HoDl - 0.221834354850636 *
   NonDimBoundaryLayerThickness - 0.664790011747447 * InletMachNumber +
  0.000266937968596887 * HeadWind - 0.00179140669537652 * URollingGround;
UzofVortexatFanFaceAppended = AppendTo[UzofVortexatFanFaceAppended,
  UzofVortexatFanFace];
Clear[sol2];
PRvortexonfanfaceAppended =
 AppendTo[PRvortexonfanfaceAppended, PRvortexonfanface];

(* Force System Differential Equations solved Symbolically *)
eqns =
```

Aspiration Simulation Code – 1.33 mm Steel Debris (continued)


```
 {((1 - 1 / h * (PRvortexonfanface - PRvortexonground) * y[t] - PRvortexonground) *
        Pambsealevel) * A0 * y'[t] / ((x'[t])^2 + (y'[t])^2)^0.5 - mimpactor * g -
    0.5 * Cdsphere * Rhosealevel * A0 * (y'[t])^2 == mimpactor * y''[t],
   ((1 - 1 / h * (PRvortexonfanface - PRvortexonground) * y[t] - PRvortexonground) *
        Pambsealevel - Rhosealevel * g * y[t]) * A0 *
     x'[t] / ((x'[t])^2 + (y'[t])^2)^0.5 - 0.5 * Cdsphere * Rhosealevel *
     A0 * (x'[t])^2 == mimpactor * x''[t], y[0] == 0,
  y'[0] == 2 * h * a / Tf, x[0] == 0, x'[0] == 2 * FodGroundDisttoFan * b / Tf,
  WhenEvent[Abs[x[t] - FodGroundDisttoFan] ≤ 0.000001, "StopIntegration"]};
sol2 = Quiet[ParametricNDSolve[eqns, {x, y}, {t, 0, 0.025},
    {Tf, a, b}, Method → "Extrapolation"]];
X1 = x[Tf, a, b] /. sol2;
Y1 = y[Tf, a, b] /. sol2;

(* Symbolically Force System Differential Equations Parametric Solution is
  solved Numerically by imposing fixed and loose physical constraints -
 the particle arrives at center of vortex at the fan face,
it arrives between ground clearance height and shaft and the
 particle cannot travel faster then the inlet flow velocity *)
sol3 = Quiet[FindMinimum[{(X1[t] - FodGroundDisttoFan)^2 +
      (Y1[t] - UzofVortexatFanFace)^2, (h + Dinlet / 2 - Dshaft / 2) ≥ Y1[t] ≥ h &&
      X1'[t] ≤ InletMachNumber * (1.4 * 287 * InletStaticTemp)^0.5 &&
      Y1'[t] ≤ InletMachNumber * (1.4 * 287 * InletStaticTemp)^0.5 && 0.01 > Tf >
       (FodGroundDisttoFan / (InletMachNumber * (1.4 * 287 * InletStaticTemp)^0.5) /
         2) && 0 < a < 1 && 0 < b < 1 && 0.01 > t > (FodGroundDisttoFan /
          (InletMachNumber * (1.4 * 287 * InletStaticTemp)^0.5) / 2) &&
      Element[Tf | a | b | t, Reals]}, {Tf, a, b, t}, Gradient -> "FiniteDifference",
   Method → "InteriorPoint", MaxIterations → 250]];

Tfx = Tf /. Extract[sol3, {2}];
ax = a /. Extract[sol3, {2}];
bx = b /. Extract[sol3, {2}];
tx = t /. Extract[sol3, {2}];
XX1 = x[Tfx, ax, bx] /. sol2;
YY1 = y[Tfx, ax, bx] /. sol2;
TfxAppended = AppendTo[TfxAppended, Tfx];
tx1 = AppendTo[tx1, tx];
ax1 = AppendTo[ax1, ax];
bx1 = AppendTo[bx1, bx];
XX1 = x[Tfx, ax, bx] /. sol2;
YY1 = y[Tfx, ax, bx] /. sol2;

(* Added Displacement and Velocity Plots *)
```

Aspiration Simulation Code – 1.33 mm Steel Debris (continued)


```
Print[Plot[XX1[t] /. sol2, {t, 0, tx},
  PlotLabel → "Aspirated Particle Horizontal Displacement",
  AxesLabel → {"t (s)", "Ux (m)"}]];
Print[Plot[YY1[t] /. sol2, {t, 0, tx},
  PlotLabel → "Aspirated Particle Vertical Displacement",
  AxesLabel → {"t (s)", "Uz (m)"}]];
Print[Plot[XX1'[t] /. sol2, {t, 0, tx}, PlotLabel → "Aspirated Particle Velocity",
  AxesLabel → {"t (s)", "Vx (m/s)"}]];
Print[ParametricPlot[{XX1[t] /. sol2, YY1[t] /. sol2}, {t, 0, tx},
  Mesh → True, PlotLabel → "Trajectory of Aspirated Particle
     from Stagnation Point to Vortex Center at Fan Face",
  GridLines → Automatic, AxesLabel → {"Ux(m)", "Uz(m)"}]];
Print[Plot[YY1'[t] /. sol2, {t, 0, tx}, PlotLabel → Vz,
  AxesLabel → {"t (s)", "Vz (m/s)"}]];

UxFanFace = AppendTo[UxFanFace, XX1[tx] /. sol2];
UyFanFace = AppendTo[UyFanFace, YY1[tx] /. sol2];
VxFanFace = AppendTo[VxFanFace, XX1'[tx] /. sol2];
VyFanFace = AppendTo[VyFanFace, YY1'[tx] /. sol2];
VortexonGroundPRatio = AppendTo[VortexonGroundPRatio, PRvortexonground];
NonDimensionalVortexStrength =
 AppendTo[NonDimensionalVortexStrength, NonDimVortexStrength];
VortexonFanFacePRatio = AppendTo[VortexonFanFacePRatio, PRvortexonfanface];
acoeffcientforVyInitial = AppendTo[acoeffcientforVyInitial, ax];
bcoeffcientforVxInitial = AppendTo[bcoeffcientforVxInitial, bx],

{k, StartingSampleSize, MaxSampleSize, SampleSizeIncrement}];

Print["Time to reach Fan Face: ", tx1];

Print["Tfx Appended: ", TfxAppended];
Print["UxFanFace: ", UxFanFace];
Print["UyFanFace: ", UyFanFace];
Print["VxFanFace: ", VxFanFace];
Print["VyFanFace: ", VyFanFace];
Print["UzofVortexatFanFaceAppended: ", UzofVortexatFanFaceAppended];

Print["VortexonGroundPRatio: ", VortexonGroundPRatio];
Print["NonDimensionalVortexStrength: ", NonDimensionalVortexStrength];
Print["VortexonFanFacePRatio: ", VortexonFanFacePRatio];
```

Aspiration Simulation Code – 1.33 mm Steel Debris (end)


```
 Print["acoeffcientforVyInitial: ", acoeffcientforVyInitial];
 Print["bcoeffcientforVxInitial: ", bcoeffcientforVxInitial];

 SteelParticleFullFactorialOutput = Join[SteelParticleFullFactorialOutput,
   Transpose[{UyFanFace, VxFanFace, VyFanFace}]];


 TestofTime =
  (FodGroundDisttoFan / (InletMachNumber * (1.4 * 287 * InletStaticTemp) ^0.5));
 InletVelocity = (InletMachNumber * (1.4 * 287 * InletStaticTemp) ^0.5);
 Print["test of time constraint: ", TestofTime];
 Print["test of Inlet Velocity: ", InletVelocity],
 {i, 1, 1, 1}]
SteelParticleFullFactorialOutput
(*Export[
 "C:\\Users\\g210029\\GeorgiaTech\\Research\\MyResearch\\MathematicaRuns\\FOD
   Inlet Velocity
   Model_Steel133mm_4Factors_NewModelUy_NoBoundaryLayer_Output.csv",
 SteelParticleFullFactorialOutput,"CSV"]*)

ListPlot[Transpose[{data1[[All, 5]], soln1["FitResiduals"]}], Frame → True,
 Filling → 0, FrameLabel → {"Non Dimensional Vortex Strength", "residual"},
 PlotLabel → "Residual vs. Predictor for NonDimVortexStrength"]
Print[soln1[{"ParameterTable", "RSquared"}]]
ListPlot[Transpose[{data2[[All, 5]], soln2["FitResiduals"]}],
 Frame → True, Filling → 0,
 FrameLabel → {"Vortex on Ground Stagnation Point Pressure Ratio", "residual"},
 PlotLabel → "Residual vs. Predictor for PRvortexonground"]
Print[soln2[{"ParameterTable", "RSquared"}]]
ListPlot[Transpose[{data3[[All, 5]], soln3["FitResiduals"]}],
 Frame → True, Filling → 0,
 FrameLabel → {"Vortex on Fan Face Stagnation Point Pressure Ratio", "residual"},
 PlotLabel → "Residual vs. Predictor for PRvortexonfanface"]
Print[soln3[{"ParameterTable", "RSquared"}]]
```

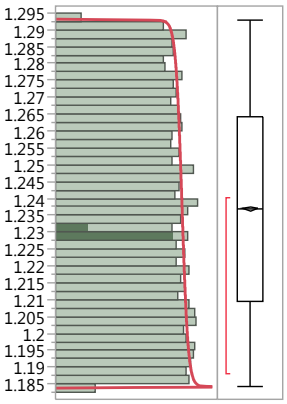
APPENDIX D: MEANLINE ANALYSIS SIMULATION & DISTRIBUTIONS

The code for the Meanline Analysis simulation is attached as a file. The simulation is in the form of a large, multi-sheet Excel spreadsheet, it is contained in the link below. The Beta distributions that were generated from 10,000 runs each for flow density and velocity (axial and tangential) across the engine stations and for inlet static temperature are valid in the bound of inlet Mach numbers that ranged uniformly from 0.45 to 0.5. The distributions are presented in table format in this appendix along with quantiles, summary statistics and parameter estimates of the fitted Beta distributions.

Meanline Analysis Spreadsheet.xlsx

Table D.1: Fan Inlet & Fan Outlet/Stator Inlet Flow Density (kg/m3)

FanInletStaticDensity



Beta(0.97771,1.02657,1.18432,0.10868)

Quantiles

100.0%	maximum	1.293
99.5%		1.29244
97.5%		1.28995
90.0%		1.28134
75.0%	quartile	1.26418
50.0%	median	1.23678
25.0%	quartile	1.2096
10.0%		1.19446
2.5%		1.18685
0.5%		1.18477
0.0%	minimum	1.18432

Summary Statistics

Mean	1.2371697
Std Dev	0.0313663
Std Err Mean	0.0003137
Upper 95% Mea	1.2377845
Lower 95% Mean	1.2365549
N	10000

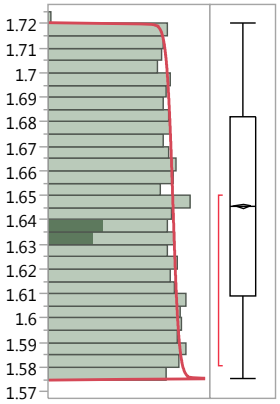
Fitted Beta

Parameter Estimates

Type	Parameter	Estimate	Lower 95%	Upper 95%
Shape	α	0.9777079	0.952978	1.002893
Shape	β	1.0265707	1.0003498	1.0532778
Threshold	θ	1.184324		
Scale	σ	0.1086799		

-2log(Likelihood) = -44397.6365096824

FanOutletStatorInletStaticDensity



Beta(0.97771,1.02657,1.57555,0.14458)

Quantiles

100.0%	maximum	1.72013
99.5%		1.71938
97.5%		1.71607
90.0%		1.70462
75.0%	quartile	1.68178
50.0%	median	1.64533
25.0%	quartile	1.60917
10.0%		1.58904
2.5%		1.57891
0.5%		1.57614
0.0%	minimum	1.57555

Summary Statistics

Mean	1.6458517
Std Dev	0.0417277
Std Err Mean	0.0004173
Upper 95% Mea	1.6466697
Lower 95% Mean	1.6450338
N	10000

Fitted Beta

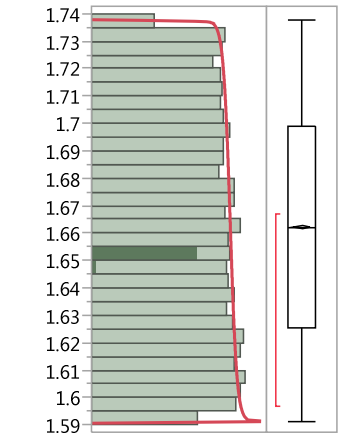
Parameter Estimates

Type	Parameter	Estimate	Lower 95%	Upper 95%
Shape	α	0.9777081	0.9529782	1.0028931
Shape	β	1.0265708	1.0003499	1.0532779
Threshold	θ	1.5755491		
Scale	σ	0.1445809		

-2log(Likelihood) = -38690.1429203161

Table D.2: Stator Outlet/IGV Inlet & IGV Outlet/Rotor Inlet Flow Density (kg/m3)

StatorOutletIgvInletStaticDensity



Beta(0.97771,1.02657,1.59146,0.14604)

Quantiles

100.0%	maximum	1.73751
99.5%		1.73675
97.5%		1.7334
90.0%		1.72183
75.0%	quartile	1.69877
50.0%	median	1.66195
25.0%	quartile	1.62542
10.0%		1.60509
2.5%		1.59486
0.5%		1.59206
0.0%	minimum	1.59146

Summary Statistics

Mean	1.6624765
Std Dev	0.0421492
Std Err Mean	0.0004215
Upper 95% Mea	1.6633027
Lower 95% Mean	1.6616503
N	10000

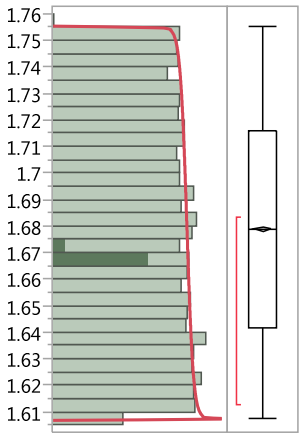
Fitted Beta

Parameter Estimates

Type	Parameter	Estimate	Lower 95%	Upper 95%
Shape	α	0.977708	0.9529781	1.002893
Shape	β	1.0265707	1.0003499	1.0532778
Threshold	θ	1.5914638		
Scale	σ	0.1460413		

-2log(Likelihood) = -38489.1765262257

IgvOutletRotorInletStaticDensity



Beta(0.97771,1.02657,1.60754,0.14752)

Quantiles

100.0%	maximum	1.75506
99.5%		1.75429
97.5%		1.75091
90.0%		1.73923
75.0%	quartile	1.71593
50.0%	median	1.67874
25.0%	quartile	1.64184
10.0%		1.6213
2.5%		1.61097
0.5%		1.60814
0.0%	minimum	1.60754

Summary Statistics

Mean	1.6792692
Std Dev	0.042575
Std Err Mean	0.0004257
Upper 95% Mea	1.6801038
Lower 95% Mean	1.6784346
N	10000

Fitted Beta

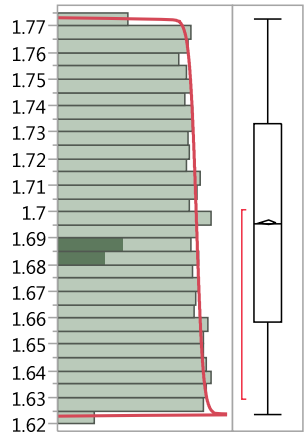
Parameter Estimates

Type	Parameter	Estimate	Lower 95%	Upper 95%
Shape	α	0.9777081	0.9529781	1.0028931
Shape	β	1.0265708	1.0003499	1.0532778
Threshold	θ	1.6075392		
Scale	σ	0.1475165		

-2log(Likelihood) = -38288.2099347251

Table D.3: Rotor Outlet Flow Density (kg/m3) & Inlet Meridional Velocity (m/s)

RotorOutletStatorInletStaticDensity



Beta(0.97771,1.02657,1.62378,0.14901)

Quantiles

100.0%	maximum	1.77278
99.5%		1.77201
97.5%		1.7686
90.0%		1.75679
75.0%	quartile	1.73326
50.0%	median	1.69569
25.0%	quartile	1.65843
10.0%		1.63768
2.5%		1.62724
0.5%		1.62438
0.0%	minimum	1.62378

Summary Statistics

Mean	1.6962315
Std Dev	0.043005
Std Err Mean	0.0004301
Upper 95% Mea	1.6970745
Lower 95% Mean	1.6953885
N	10000

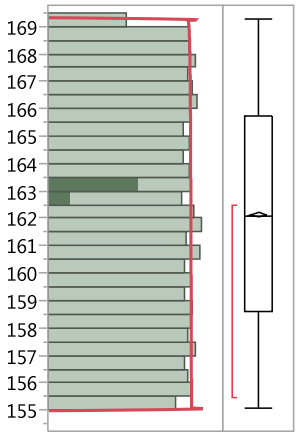
Fitted Beta

Parameter Estimates

Type	Parameter	Estimate	Lower 95%	Upper 95%
Shape	α	0.977708	0.9529781	1.002893
Shape	β	1.0265708	1.0003499	1.0532779
Threshold	θ	1.6237769		
Scale	σ	0.1490065		

-2log(Likelihood) = -38087.2434903091

FanInletCm



Beta(0.99941,1.00407,155.063,14.2294)

Quantiles

100.0%	maximum	169.292
99.5%		169.229
97.5%		168.931
90.0%		167.855
75.0%	quartile	165.755
50.0%	median	162.112
25.0%	quartile	158.598
10.0%		156.474
2.5%		155.43
0.5%		155.13
0.0%	minimum	155.063

Summary Statistics

Mean	162.1647
Std Dev	4.1085338
Std Err Mean	0.0410853
Upper 95% Mea	162.24524
Lower 95% Mean	162.08417
N	10000

Fitted Beta

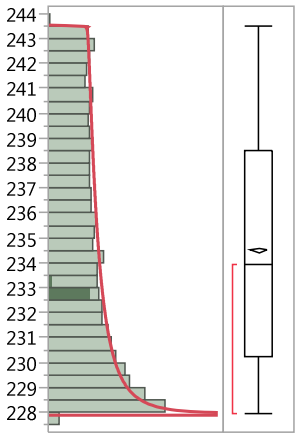
Parameter Estimates

Type	Parameter	Estimate	Lower 95%	Upper 95%
Shape	α	0.9994096	0.9740202	1.0252679
Shape	β	1.0040655	0.978534	1.0300688
Threshold	θ	155.06251		
Scale	σ	14.229372		

-2log(Likelihood) = 53095.3458949987

Table D.4: Fan Outlet/Stator Inlet Tangential Velocities (m/s)

WTheta2



Beta(0.75362,1.01229,227.965,15.5477)

Quantiles

100.0%	maximum	243.513
99.5%		243.418
97.5%		242.994
90.0%		241.51
75.0%	quartile	238.517
50.0%	median	233.952
25.0%	quartile	230.21
10.0%		228.691
2.5%		228.122
0.5%		227.991
0.0%	minimum	227.965

Summary Statistics

Mean	234.52184
Std Dev	4.6750481
Std Err Mean	0.0467505
Upper 95% Mea	234.61348
Lower 95% Mean	234.4302
N	10000

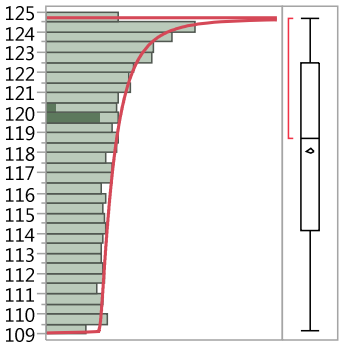
Fitted Beta

Parameter Estimates

Type	Parameter	Estimate	Lower 95%	Upper 95%
Shape	α	0.7536224	0.7351388	0.7724384
Shape	β	1.0122938	0.985818	1.0392728
Threshold	θ	227.96522		
Scale	σ	15.547713		

-2log(Likelihood) = 53938.9987426737

CTheta2



Beta(1.01229,0.75362,109.15,15.5477)

Quantiles

100.0%	maximum	124.698
99.5%		124.672
97.5%		124.541
90.0%		123.972
75.0%	quartile	122.453
50.0%	median	118.712
25.0%	quartile	114.146
10.0%		111.153
2.5%		109.669
0.5%		109.245
0.0%	minimum	109.15

Summary Statistics

Mean	118.14127
Std Dev	4.6750481
Std Err Mean	0.0467505
Upper 95% Mea	118.23291
Lower 95% Mean	118.04963
N	10000

Fitted Beta

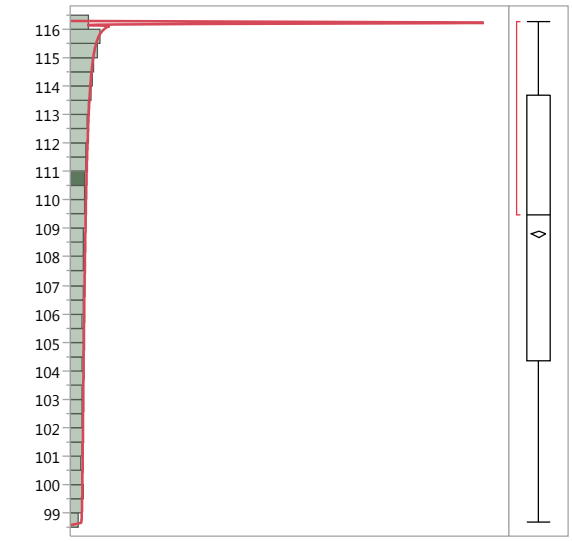
Parameter Estimates

Type	Parameter	Estimate	Lower 95%	Upper 95%
Shape	α	1.0122937	0.9858179	1.0392727
Shape	β	0.7536223	0.7351387	0.7724383
Threshold	θ	109.15017		
Scale	σ	15.547713		

-2log(Likelihood) = 53938.9981363443

Table D.5: Stator Outlet/IGV Inlet Tangential Velocities (m/s)

CTheta3



Beta(1.02509,0.76497,98.6535,17.6066)

Quantiles

100.0%	maximum	116.26
99.5%		116.226
97.5%		116.072
90.0%		115.411
75.0%	quartile	113.68
50.0%	median	109.467
25.0%	quartile	104.324
10.0%		100.933
2.5%		99.2628
0.5%		98.7727
0.0%	minimum	98.6535

Summary Statistics

Mean	108.82379
Std Dev	5.2763926
Std Err Mean	0.0527666
Upper 95% Mea	108.92722
Lower 95% Mean	108.72035
N	9999

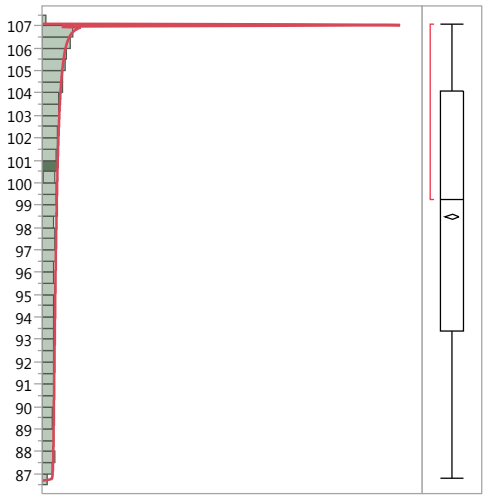
Fitted Beta

Parameter Estimates

Type	Parameter	Estimate	Lower 95%	Upper 95%
Shape	α	1.0250914	0.9982796	1.0524125
Shape	β	0.7649703	0.7461886	0.7840901
Threshold	θ	98.653505		
Scale	σ	17.606637		

-2log(Likelihood) = 56468.3888263204

CTheta4



Beta(1.03423,0.77138,86.7865,20.2726)

Quantiles

100.0%	maximum	107.059
99.5%		107.019
97.5%		106.837
90.0%		106.063
75.0%	quartile	104.07
50.0%	median	99.2383
25.0%	quartile	93.3579
10.0%		89.4424
2.5%		87.4973
0.5%		86.9259
0.0%	minimum	86.7865

Summary Statistics

Mean	98.502659
Std Dev	6.0554419
Std Err Mean	0.0605544
Upper 95% Mea	98.621358
Lower 95% Mean	98.383961
N	10000

Fitted Beta

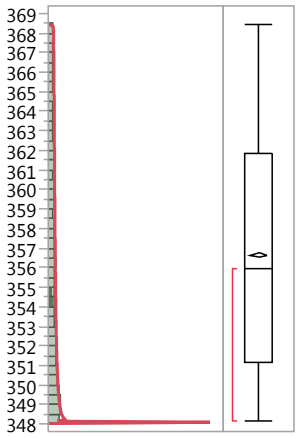
Parameter Estimates

Type	Parameter	Estimate	Lower 95%	Upper 95%
Shape	α	1.0342328	1.0071765	1.0618029
Shape	β	0.7713807	0.7524341	0.7906683
Threshold	θ	86.786547		
Scale	σ	20.27261		

-2log(Likelihood) = 59311.1854415389

Table D.6: Outlet/Rotor Inlet & Rotor Outlet Tangential Velocities (m/s)

WTheta4



Beta(0.77138,1.03423,348.133,20.2726)

Quantiles

100.0%	maximum	368.405
99.5%		368.266
97.5%		367.694
90.0%		365.749
75.0%	quartile	361.834
50.0%	median	355.953
25.0%	quartile	351.122
10.0%		349.129
2.5%		348.355
0.5%		348.173
0.0%	minimum	348.133

Summary Statistics

Mean	356.689
Std Dev	6.0554419
Std Err Mean	0.0605544
Upper 95% Mea	356.8077
Lower 95% Mean	356.57031
N	10000

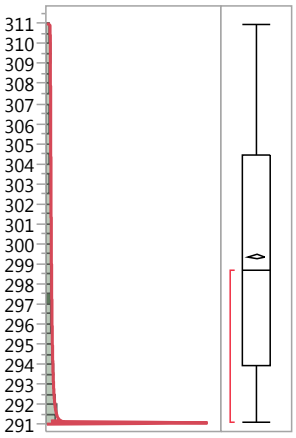
Fitted Beta

Parameter Estimates

Type	Parameter	Estimate	Lower 95%	Upper 95%
Shape	α	0.7713806	0.7524341	0.7906683
Shape	β	1.0342328	1.0071765	1.0618029
Threshold	θ	348.13251		
Scale	σ	20.27261		

-2log(Likelihood) = 59311.1851804423

WTheta5



Beta(0.7555,1.03052,291.104,19.8739)

Quantiles

100.0%	maximum	310.978
99.5%		310.84
97.5%		310.273
90.0%		308.346
75.0%	quartile	304.462
50.0%	median	298.644
25.0%	quartile	293.922
10.0%		292.022
2.5%		291.306
0.5%		291.14
0.0%	minimum	291.104

Summary Statistics

Mean	299.40862
Std Dev	5.9531143
Std Err Mean	0.0595311
Upper 95% Mea	299.52531
Lower 95% Mean	299.29193
N	10000

Fitted Beta

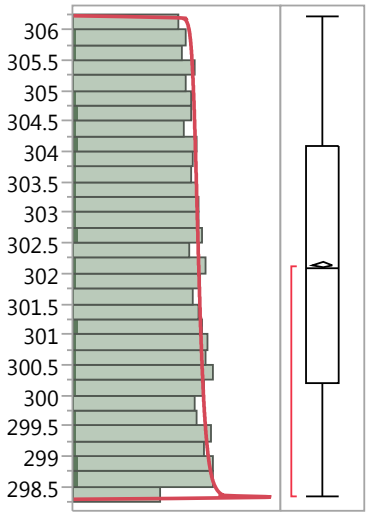
Parameter Estimates

Type	Parameter	Estimate	Lower 95%	Upper 95%
Shape	α	0.7554985	0.7369848	0.7743449
Shape	β	1.0305152	1.0035057	1.0580388
Threshold	θ	291.1042		
Scale	σ	19.873919		

-2log(Likelihood) = 58791.1652979765

Table D.7: Fan Inlet Static Temperature (°K)

FanInletStaticTemperature



Beta(0.96348,1.01802,298.343,7.86775)

Quantiles

100.0%	maximum	306.21
99.5%		306.173
97.5%		306.018
90.0%		305.37
75.0%	quartile	304.078
50.0%	median	302.088
25.0%	quartile	300.198
10.0%		299.05
2.5%		298.519
0.5%		298.377
0.0%	minimum	298.343

Summary Statistics

Mean	302.15354
Std Dev	2.2725101
Std Err Mean	0.0227251
Upper 95% Mea	302.19809
Lower 95% Mean	302.10899
N	10000

Fitted Beta

Parameter Estimates

Type	Parameter	Estimate	Lower 95%	Upper 95%
Shape	α	0.9634764	0.9391342	0.9882663
Shape	β	1.0180206	0.9920129	1.0445109
Threshold	θ	298.34267		
Scale	σ	7.8677468		

-2log(Likelihood) = 41221.4100690775

APPENDIX E: IN-ENGINE PARTICLE KINETICS SIMULATION

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris

```
ClearSystemCache
Off[Power::infy]
"Statistics`NonlinearFit`"
AppendTo[$Path, ToFileName[{$HomeDirectory, "MathematicaRuns"}]];

(* Input .CSV file contains Full Factorial (81) and 1,000 Latin Hypercube, Space Filling DOE *)
RadialParticleMotionInputNoBoundaryLayer = Import[
  "ParticleMotionNoBoundaryLayerHeadWindandUGEqual32mmSteelInput_FullFactorialSpaceFilling1081Runs
    .csv", "CSV"]
with[{g = 9.8, rhosteel = 7850, Cdsphere = 0.47,
   Pambsealevel = 101 325, Rhosealevel = 1.225, ρTi6Al4v = 4430}];

(* Vectors are Initialized *)
SteelParticleFullFactorialOutput = {};
MeanVSuctionSideAppended = {};
MeanVPressureSideAppended = {};
STDVSuctionSideAppended = {};
STDVPressureSideAppended = {};
MeanRimpactAppendedSuctionSiderotor = {};
MeanRimpactAppendedPressureSiderotor = {};
STDRimpactAppendedSuctionSiderotor = {};
STDRimpactAppendedPressureSiderotor = {};
RimpactVsXpctAppendedSuctionSiderotor = {};
RimpactVsXpctAppendedPressureSiderotor = {};
VSuctionSidevsChordLocationAppended = {};
VPressureSidevsChordLocationAppended = {};
VSuctionSidevsChordLocationAppendedTotal = {};
VPressureSidevsChordLocationAppendedTotal = {};
RimpactAppendedSuctionSiderotorTotal = {};
RimpactAppendedPressureSiderotorTotal = {};
XXPercentChordLocationPressureSiderotorTotal = {};
XXPercentChordLocationSuctionSiderotorTotal = {};
MeanXXPercentChordLocationSuctionSiderotor = {};
MeanXXPercentChordLocationPressureSiderotor = {};
STDXXPercentChordLocationSuctionSiderotor = {};
STDXXPercentChordLocationPressureSiderotor = {};
RIMPACTRotorAppend = {};
XpercentChordAppend = {};
ImpactSideRotorAppend = {};
ImpactVelocityatRotor = {};
StressatImpactSiteAppend = {};
DepthofPenetrationAppend = {};
DiameterofPenetrationAppend = {};
ChordatImpactAppend = {};
BetaCFCAppend = {};
BetaWAppend = {};
BetaFFSAppend = {};
BetaCompoundedAppend = {};
LocalThicknessatImpactAppended = {};
NimpactAppended = {};
```

Printed by Wolfram Mathematica Student Edition

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)

```
2 |
(* Counters are initialized *)
NrunsCount = 0;
NfanstrikesTotal = 0;
NstatorstrikesTotal = 0;
NigvstrikesTotal = 0;
NrotorstrikesTotal = 0;
NshaftstrikesTotal = 0;
NbypassexitTotal = 0;
NtaperedfanhousingstrikesTotal = 0;

(* Gages the length of the DOE input file, a .CSV file *)
InputFileLength = Length[RadialParticleMotionInputNoBoundaryLayer];

(* FOD particle sizing *)
rfod = 0.0032 / 2;
Dfod = rfod * 2;
Print["Radius of FOD (m):", rfod]

(* Mass of FOD particle *)
m = (4 / 3 * Pi * rfod^3) * rhosteel;

(* Area of FOD particle *)
A0 = Pi * rfod^2;

(* Geometric engine parameters from Meanline Analysis turbomachinery sizing at Mach 0.475 *)
Rrms = 0.328315858; (*Fan Radius at rms, meanline radius*)
Crms = 0.0759; (*Fan Chord at Rrms Reference*)
Cstatorrms = 0.07042; (*Stator Chord at Rrms Reference*)
Cigvrms = 0.05809; (*igv Chord at Rrms Reference*)
Cistrotorrms = 0.02364; (*1st Rotor Chord at Rrms Reference*)
λtip = 62.78 + 5.56; (*Fan Twist Angle at Tip of Fan*)
λstator = 29.21777; (*Stator Twist Angle, Stagger Angle*)
λigv = 17.48147; (*Stator Twist Angle, Stagger Angle*)
λistrotortip = 65.47 + 4.90; (*1st Rotor Twist Angle at tip*)
Dinlet = 0.88;
Dfanoutlet = 0.85493;
(*Fan Duct diamter at fan outlet, meanline model calculates this*)
Dstatoroutlet = 0.85493;
(*Stator Duct diamter at stator outlet, meanline model calculates this*)
Digvoutlet = 0.77123;
(*IGV Duct diamter at igv outlet, meanline model calculates this*)
Distrotoroutlet = 0.77123;
(*Rotor Duct diamter at rotor outlet, meanline model calculates this*)
Rfanrms = (0.319863252 + 0.336768463) / 2;
(*Average Rrms at Fan, must come from Meanline spreadsheet*)
Rstatorrms = (0.336768463 + 0.336768463) / 2;
(*Average Rrms at Stator, must come from Meanline spreadsheet*)
Rigvrms = (0.310482871 + 0.310482871) / 2;
(*Average Rrms at igv, must come from Meanline spreadsheet*)
Rrotorrms = (0.310482871 + 0.310482871) / 2;
(*Average Rrms at 1st Rotor, must come from Meanline spreadsheet*)
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)


```
(* Fixed shaft geometry *)
Rshaft = 0.21;
Dshaft = 0.42;

(* Stator and IGV airfoils spacing,
can be dialed up to reduce process time if Stator and IGV are not in the path of particle *)
StatorAirfoilSpacing = 10 * Dfod;
IGVAirfoilSpacing = 10 * Dfod;

(* Assumed Chord Tip to Root ratio and Twist at Tip to Root ratio*)
CtiptoCrootRatio = 1.01;
TwistTiptoRoot = 0.2;

(* Angular velocities of the Low Pressure Compressor
 (the fan) and of the High Pressure Compressor spools *)
ωfan = 10 000 * 2 * Pi / 60;
Print["Fan (N1) Speed (rads/s): ", ωfan]
ωrotor = 14 000 * 2 * Pi / 60;
Print["HP Compressor (N2) Speed (rads/s): ", ωrotor]

(* Chord scaling relationships *)
HtoCrmsRatio = (Dinlet / 2 + Rfanrms * (CtiptoCrootRatio - 1) - CtiptoCrootRatio * Rshaft) /
    (Crms * (CtiptoCrootRatio - 1));
HtoCrmsRatio1strotor = (Distrotoroutlet / 2 + Rrotorrms * (CtiptoCrootRatio - 1) -
      CtiptoCrootRatio * Rshaft) / (Cistrotorrms * (CtiptoCrootRatio - 1));
Cistrotortip = (Distrotoroutlet / 2 - Rrotorrms) / HtoCrmsRatio1strotor + Cistrotorrms;
Cfantip = (Dinlet / 2 - Rfanrms) / HtoCrmsRatio + Crms; (*1st Rotor Max Chord at Tip*)
Cfanroot = (Dshaft / 2 - Rfanrms) / HtoCrmsRatio + Crms;
Print["HtoCrmsRatio1strotor: ", HtoCrmsRatio1strotor]
Print["Cistrotortip: ", Cistrotortip]
Print["HtoCrmsRatio: ", HtoCrmsRatio]
Print["Cfantip: ", Cfantip]
Print["Cfanroot: ", Cfanroot]
Print["CtiptoCrootRatio", CtiptoCrootRatio]

(* NACA 65-210 airfoil outermold line node coordinates scaled
  to chord and twisted to stagger angle at rrms for fan and 1st HPC rotor,
stator and IGV are fixed in stagger and chord *)
Xairfoil = {0.0347231411378479, 0.0325719368641676, 0.0303646997900155,
   0.0281391964406781, 0.0259150045416217, 0.0237093293049559, 0.0215343417911906,
   0.0194014992291716, 0.0173212946288752, 0.0153051852191473, 0.0133683896678353,
   0.0115301384689939, 0.00978234986368515, 0.00811657549271868, 0.00652508082856798,
   0.00500987178433663, 0.0035794160880562 5, 0.00224484310600242, 0.00103554847963596,
   0.00049556562624117, 0.0000190803447979225, -0.000350006887049441, -0.000453481118659142,
   -0.000438995990691066, -0.000401856306970329, 0, 0.000681578320190082,
   0.000865329950336568, 0.00117708953504567, 0.00183502898279138, 0.00302657437262372,
   0.00414042584805002, 0.0052110325709724, 0.00726342273613551, 0.00923509074839601,
   0.0111432805575461, 0.0129964211539352, 0.0147989297017051, 0.0165501117380333,
   0.0182422327353932, 0.0198654941466311, 0.0214118142128093, 0.0229004234279824,
   0.024347890647961, 0.0257656731015909, 0.0271669641747748, 0.0285672716602481,
```

```
4 |
    0.0299814669944469, 0.0314389070811496, 0.0329753719013804, 0.0347231411378479};
Yairfoil = {0.0675093981587262, 0.0643593575044095, 0.061238137057237 3,
    0.0581228970194231, 0.0550035676800371, 0.0518704459284018, 0.048718979018487,
    0.0455415665969246, 0.0423353727059001, 0.0390927969920456, 0.0358085728776668,
    0.0324719555578522, 0.0290888092289129, 0.0256626255738039, 0.0221999438828177,
    0.0186980250423458, 0.0151550747888077, 0.0115645150942285, 0.0079137875212 7487,
    0.00605729810137677, 0.0041715637675582, 0.00223571189745412, 0.00123121045116031,
    0.000804597899483265, 0.00057804840790 9434, 0, 0.000131768714815676,
    0.000256655470490616, 0.000530832024842836, 0.00126892809551499, 0.00279882630030562,
    0.00436356385928111, 0.00594712938983572, 0.00915567783310484, 0.0124014768815973,
    0.0156782182633848, 0.0189807128761378, 0.0223092498447441, 0.0256624789812407,
    0.0290469395759202, 0.0324668177745603, 0.0359279777734721, 0.0394196747677441,
    0.0429359477168592, 0.0464691962674495, 0.050015195536055, 0.0535642617808394,
    0.0571104533301269, 0.0606378169078353, 0.0641279492491615, 0.0675093981587262};
Xrotorairfoil = {0.00981574190413467, 0.00919256673891168, 0.00855154226704516,
    0.00790474886832068, 0.00725842304746913, 0.00661805799614051, 0.00598750591125653,
    0.00537044141194999, 0.0047701696968998, 0.00419036538523878, 0.00363583903885675,
    0.00311272900506554, 0.0026184359241687, 0.00215028101609994, 0.00170576307858185,
    0.00128554600432542, 0.000892289836347787, 0.000529552299313506, 0.000206660044745105,
    0.000065781714355659 9, -0.000054919212728 6477, -0.000141483100884362, -0.000159015029637212,
    -0.000148286105274138, -0.000133429018392326, 0, 0.000210081248706528,
    0.000265414969618156, 0.00035838747336449, 0.000552270894026119, 0.000900580610386563,
    0.00122421557085836, 0.00153412468464316, 0.00212571043165953, 0.00269164432076994,
    0.00323738206952603, 0.00376562119504327, 0.00427773142901516, 0.0047735164566036,
    0.0052504750755312, 0.00570549840195712, 0.00613598707618489, 0.00654807944552637,
    0.00694701617564591, 0.00733647188567632, 0.00772061198184561, 0.0081044013946102,
    0.00849255252844089, 0.00889442950895858, 0.00932143008618864, 0.00981574190413467};
Yrotorairfoil = {0.0215051887271443, 0.0204939939318526, 0.0194909462023413,
    0.018489492200757, 0.0174867853619474, 0.0164800585411303, 0.0154680731659563,
    0.0144486322991084, 0.0134210068923816, 0.012383000008 4587, 0.0113331954972017,
    0.0102685317415711, 0.00919071489178887, 0.00810070778499637, 0.00700043162330291,
    0.00588906367325777, 0.00476616935891332, 0.00362986495225859, 0.00247667299735564,
    0.00189133877352294, 0.00129783427468752, 0.00069030752980528, 0.000376350050659951,
    0.000243864441192344, 0.000173938497157941, 0, 0.000050929132017 6368,
    0.000092455428380 6091, 0.000182285355193504, 0.000421466466435314, 0.000914720086861037,
    0.00141767702689576, 0.00192585951936197, 0.00295381155191435, 0.00399217274305821,
    0.00503923247039837, 0.00609349963560003, 0.00715512860722979, 0.00822368928151311,
    0.00930110286055934, 0.0103885284963478, 0.0114876723391005, 0.0125954727717781,
    0.0137103173294044, 0.014830009074662, 0.0159534263296702, 0.0170777832125972,
    0.0182014485214179, 0.0193198882778071, 0.0204279001385557, 0.0215051887271443};
Xstatorairfoil = {0.0614615388913439, 0.058183255817534, 0.054876441727954,
    0.0515585754973694, 0.048239625793941, 0.0449279154289067, 0.0416305173350694,
    0.0383523897308546, 0.0351001889229204, 0.031878873129692, 0.0286975048880734,
    0.0255654382508361, 0.0224794337354447, 0.0194347517603748, 0.0164287674584797,
    0.0134616265876935, 0.0105389542131402, 0.00766597942145638, 0.00485898344028269,
    0.00348950609648479, 0.00215411195461432, 0.000876030064838182, 0.000280895232104223,
    0.000073305802907151 7, -0.000014171244567 0735, 0, 0.000594411915685101,
    0.000800492675549431, 0.00118208134409719, 0.00206917292953008, 0.00377479371805306,
    0.00543822713031 17, 0.00707788975598153, 0.0103097985444205, 0.0134984158360944,
    0.0166538354094406, 0.0197799113623838, 0.0228802062769804, 0.0259534909224527,
    0.0289971404316539, 0.0320057277287037, 0.0349760133310667, 0.0379173512915146,
```

```
    0.040839491641307, 0.0437473925988692, 0.0466490854595713, 0.0495515909249063,
    0.052463356916389, 0.0553988936944605, 0.0583764198595524, 0.0614615388913439};
Ystatorairfoil = {0.0343747177954114, 0.0330430851380363, 0.0317624655579411,
    0.0304958363738569, 0.0292253737395052, 0.0279347539075963, 0.0266142159108747,
    0.0252520091779839, 0.023840560335013, 0.0223681188106057, 0.0208228089231287,
    0.0191864630791255, 0.0174677587730079, 0.0156737276422649, 0.0138133909284862,
    0.0118836026757603, 0.00987863335634302, 0.00778769076144484, 0.00558599228626528,
    0.00442792126236806, 0.00321468061140263, 0.00190762244283442, 0.00118424584111517,
    0.000847061360177415, 0.000652900025930146, 0, -0.000247691309084688,
    -0.000245394438798362, -0.000193350203002406, 0.0000397503716119508, 0.000611228093931807,
    0.00124948025674224, 0.0019244635849535, 0.00335199689065607, 0.00484972146097174,
    0.00640391776810045, 0.00800625206480838, 0.00965468251568481, 0.0113485216263738,
    0.0130967906384652, 0.0149077504202258, 0.0167900784660672, 0.0187256071825333,
    0.0207012316572126, 0.022705201318749, 0.0247274843316758, 0.0267526424323032,
    0.02876845619182, 0.0307475387859359, 0.0326573155990386, 0.0343747177954114};
Xigvairfoil = {0.0554075041972473, 0.0525363456773025, 0.0496507034757891,
    0.0467584824194186, 0.0438647430716304, 0.0409734684142596, 0.0380887331345787,
    0.0352125701218802, 0.032349084631916, 0.0295003095539784, 0.0266715716591324,
    0.0238673774357756, 0.0210865665055536, 0.0183264907671733, 0.0155865439173225,
    0.0128663158146673, 0.010169388208397, 0.00749817356065844, 0.00486166318845547,
    0.00356128364668451, 0.00227917460633812, 0.00102761235750486, 0.00042557243928605,
    0.000201335193072426, 0.0000981053516361621, 0, 0.000438519471579063,
    0.000605347063626136, 0.000922271697504621, 0.00167784803858054, 0.0031512904308408,
    0.00460186412784112, 0.00603940238904404, 0.00888919775404319, 0.0117158059878865,
    0.0145250773682016, 0.017318726348385, 0.0200992876645005, 0.022865653166464,
    0.0256172165509632, 0.02835098071233, 0.0310657852722099, 0.0337661367190005,
    0.036457710828142, 0.0391425404889266, 0.0418254289658566, 0.0445094561613499,
    0.0471993947714293, 0.0499023688173061, 0.0526276269149216, 0.0554075041972473};
Yigvairfoil = {0.0174502187572483, 0.0169247855261188, 0.016445340523473, 0.0159790493881262,
    0.0155098439438612, 0.0150231437373992, 0.0145098784974018, 0.0139597258687383,
    0.0133654523073838, 0.0127167354582077, 0.0120024632518913, 0.0112063930100298,
    0.0103360767125444, 0.00939798875774996, 0.00839985562775607, 0.00733911201801929,
    0.00621018660964064, 0.00500348562471901, 0.00369626141805429, 0.00299072547872446,
    0.00223491264428345, 0.00139371111642017, 0.000909330585703191, 0.000671833450104644,
    0.000529695910969486, 0, -0.000299786219199755, -0.000332509662869773,
    -0.000354503057789874, -0.000315084262943999, -0.000139715342032018,
    0.000096662970455444 3, 0.00036669592679491, 0.000977365008950429, 0.00165198836007886,
    0.00237779168573559, 0.00314739750818741, 0.00395855895878715, 0.00481092703315969,
    0.00571222821341338, 0.00667004504794826, 0.00769192951668511, 0.00876163894121918,
    0.00986695310214 78, 0.0109975496443403, 0.012143978723604, 0.0132925935345562,
    0.0144321075056909, 0.0155379668329735, 0.0165808055391796, 0.0174502187572483};


Do[
 ClearAll[ImpactorMaterial, aa, DepthofPenetration];
 Rimpact = 0;
 XpercentChord = 0;
 SurfaceLocator = 0;
 VSuctionSide = 0;
 VPressureSide = 0;
 Axialc = 0;
```

```
6 |
DepthofPenetration = 0;
PenetrationDiam = 0;
ChordatImpact = 0;
BetaCFC = 0;
ImpactEccentricty = 0;
AverageAirfoilThickness = 0;
G1 = 0;
G2 = 0;
G3 = 0;
G4 = 0;
BetaW = 0;
BetaFFS = 0;
BetaCompounded = 0;
LocalThicknessatImpact = 0;
Nimpact = 0;

NrunsCount = NrunsCount + Nruns;
(* Extracts engine ground clearance height from DOE input file, varies from 0.24 to 0.33 m *)
h = RadialParticleMotionInputNoBoundaryLayer[[k, 1]];
(* Extracts engine ground distance to fan face from DOE input file, varies from 0.5 to 1 m *)
GroundDistancetoFanFace = RadialParticleMotionInputNoBoundaryLayer[[k, 2]];
(* Extracts engine inlet Mach Number from DOE input file, varies from 0.45 to 0.5 *)
InletMachNumber = RadialParticleMotionInputNoBoundaryLayer[[k, 3]];
HoDi = h / Dinlet;
NonDimBoundaryLayerThickness = 0;

(* In-engine aero/thermo dynamics are linked to the inlet Mach number,
since it comes from a uniform distribution a random number
 is indexed to it to extract aero/thermo dynamic values from Beta
 distributions created from 10000 runs of the Meanline Analysis code *)
RandomNumber1 = (InletMachNumber - 0.45) / 0.05;
(* Second random number is used to
 randomly offset the locatio of fan and 1st HPC rotor blades *)
RandomNumber2 = RandomReal[];

(* Extracts Head Wind from DOE input file, varies from 9.357 to 12 m/s *)
HeadWind = RadialParticleMotionInputNoBoundaryLayer[[k, 4]];

(* For this case to match Murphy's experimental data for no boundary layer and Rolling Ground,
Headwind is equal to Rolling Ground *)
URollingGround = HeadWind;

(* Vertical location of particle at inlet/fan face, formulation is specific to particle size,
the aspiration model must be run to particle size to generate data for this fit *)
ZinitialParticleatFanFace =
 0.389538925333117 + 1.11624082304527 * h + 0.00105085125925928 * GroundDistancetoFanFace +
   (h - 0.285) * ((GroundDistancetoFanFace - 0.75) * -0.0431762864197533) +
   -0.631804631851853 * InletMachNumber + (h - 0.285) * ((InletMachNumber - 0.475) * 0.366651802469144) +
   (GroundDistancetoFanFace - 0.75) * ((InletMachNumber - 0.475) * 0.073427737777779) +
   (h - 0.285) * ((GroundDistancetoFanFace - 0.75) * ((InletMachNumber - 0.475) * -1.35165911111109)) +
   -0.00170742960440577 * HeadWind + (h - 0.285) * ((HeadWind - 10.6785) * -0.00360391156700902) +
```

```
  (GroundDistancetoFanFace - 0.75) * ((HeadWind - 10.6785) * -0.000303700088283511) +
  (h - 0.285) * ((GroundDistancetoFanFace - 0.75) * ((HeadWind - 10.6785) * -0.0727352755706903)) +
  (InletMachNumber - 0.475) * ((HeadWind - 10.6785) * 0.00839071677807183) +
  (h - 0.285) * ((InletMachNumber - 0.475) * ((HeadWind - 10.6785) * 0.435551323552084)) +
  (GroundDistancetoFanFace - 0.75) *
   ((InletMachNumber - 0.475) * ((HeadWind - 10.6785) * -0.0483523571698826)) +
  (h - 0.285) * ((GroundDistancetoFanFace - 0.75) *
      ((InletMachNumber - 0.475) * ((HeadWind - 10.6785) * -0.691514020263169))) +
  (h - 0.285) * ((h - 0.285) * -0.20830705075444) + (GroundDistancetoFanFace - 0.75) *
   ((GroundDistancetoFanFace - 0.75) * -0.00766424355555577) +
  (InletMachNumber - 0.475) * ((InletMachNumber - 0.475) * 0.0350068444443871) +
  (HeadWind - 10.6785) * ((HeadWind - 10.6785) * 0.000193334458529425);

Print[Style["ZinitialParticleatFanFace: ", FontSize → 48, Red, Bold]];
Print[Style[ZinitialParticleatFanFace, FontSize → 48, Red, Bold]];

ClearAll[r, e, t, Theta];
Off[Power::infy];

(* Ground to Fan Face vortex Aspiration model inlet velocity fit,
formulation is specific to particle size,
the aspiration model must be run to particle size to generate data for this fit *)
Vinlet = (-36.9719629605654) + 23.2174566090539 * h + 101.228604335555 * GroundDistancetoFanFace +
  (h - 0.285) * ((GroundDistancetoFanFace - 0.75) * 319.078165827162) +
  128.336327644444 * InletMachNumber + (h - 0.285) * ((InletMachNumber - 0.475) * 1493.46596469135) +
  (GroundDistancetoFanFace - 0.75) * ((InletMachNumber - 0.475) * 183.049629377773) +
  (h - 0.285) * ((GroundDistancetoFanFace - 0.75) * ((InletMachNumber - 0.475) * 6088.63611703699)) +
  -0.597310977704914 * HeadWind + (h - 0.285) * ((HeadWind - 10.6785) * -30.2703849021172) +
  (GroundDistancetoFanFace - 0.75) * ((HeadWind - 10.6785) * -1.34432020515403) +
  (h - 0.285) * ((GroundDistancetoFanFace - 0.75) * ((HeadWind - 10.6785) * 8.0141109009135)) +
  (InletMachNumber - 0.475) * ((HeadWind - 10.6785) * 0.506634279227802) +
  (h - 0.285) * ((InletMachNumber - 0.475) * ((HeadWind - 10.6785) * 999.706992054484)) +
  (GroundDistancetoFanFace - 0.75) *
   ((InletMachNumber - 0.475) * ((HeadWind - 10.6785) * 229.379758355403)) +
  (h - 0.285) * ((GroundDistancetoFanFace - 0.75) * ((InletMachNumber - 0.475) *
       ((HeadWind - 10.6785) * 5252.3554580233))) + (h - 0.285) * ((h - 0.285) * 262.432315500684) +
  (GroundDistancetoFanFace - 0.75) * ((GroundDistancetoFanFace - 0.75) * -26.2142237244445) +
  (InletMachNumber - 0.475) * ((InletMachNumber - 0.475) * -1083.38361422222) +
  (HeadWind - 10.6785) * ((HeadWind - 10.6785) * 0.983731606785603);
Print["Vinlet: ", Vinlet];

(* Locates particle radially from engine centerline axis *)
RinitialParticle = Dinlet / 2 + h - ZinitialParticleatFanFace;

nfanblades = 34;
Print["Number of Fan & Stator Blades: ", nfanblades];
nrotorblades = 38;
Print["Number of igv & 1st Rotor Blades: ", nrotorblades];
(*StatorBladeOffset=RadialParticleMotionInputNoBoundaryLayer[[k,4]]*Pi/nfanblades;*)
(*Used if StatorBladeOffset is varied*)
(*Sets Offset from Datum of In-line with Start of 1st Fan Blade at Bottom of Engine*)
```

```
8 |
  StatorBladeOffset = -0.25 * Pi / nfanblades;
  (*IGVBladeOffset=RadialParticleMotionInputNoBoundaryLayer[[k, 5]]*Pi/nrotorblades;*)
  (*Used if IGVBladeOffset is varied*)
  (*Sets Offset from Datum of In-line with Start of 1st Fan Blade at Bottom of Engine*)
  IGVBladeOffset = -0.25 * Pi / nrotorblades;

  (*Static Density at Fan Inlet, model comes from Meanline spreadsheet*)
  ρfaninletstaticdensitymin = 1.184323952;
 ρfaninletstaticdensitymax = 1.293003901;
  alphaρfaninletstaticdensity = 0.977707949073343;
  betaρfaninletstaticdensity = 1.02657069496025;
  ρfaninletstaticdensity =
   InverseBetaRegularized[RandomNumber1, alphaρfaninletstaticdensity, betaρfaninletstaticdensity] *
     (ρfaninletstaticdensitymax - ρfaninletstaticdensitymin) + ρfaninletstaticdensitymin;


  (*Static Density at Fan Outlet/Stator Inlet, comes from Meanline spreadsheet*)
  ρfanoutletstatorinletstaticdensitymin = 1.575549122;
 ρfanoutletstatorinletstaticdensitymax = 1.720130003;
  alphaρfanoutletstatorinletstaticdensity = 0.977708085009013;
  betaρfanoutletstatorinletstaticdensity = 1.02657083296519;
 ρfanoutletstatorinletstaticdensity = InverseBetaRegularized[RandomNumber1,
      alphaρfanoutletstatorinletstaticdensity, betaρfanoutletstatorinletstaticdensity] *
     (ρfanoutletstatorinletstaticdensitymax - ρfanoutletstatorinletstaticdensitymin) +
    ρfanoutletstatorinletstaticdensitymin;

  (*Static Density at Stator Outlet/IGV Inlet, comes from Meanline spreadsheet*)
  ρstatoroutletigvinletstaticdensitymin = 1.59146376;
 ρstatoroutletigvinletstaticdensitymax = 1.737505053;
  alphaρstatoroutletigvinletstaticdensity = 0.977708011523129;
  betaρstatoroutletigvinletstaticdensity = 1.02657074914944;
 ρstatoroutletigvinletstaticdensity = InverseBetaRegularized[RandomNumber1,
      alphaρstatoroutletigvinletstaticdensity, betaρstatoroutletigvinletstaticdensity] *
     (ρstatoroutletigvinletstaticdensitymax - ρstatoroutletigvinletstaticdensitymin) +
    ρstatoroutletigvinletstaticdensitymin;


  (*Static Density at IGV Outlet/Rotor Inlet, comes from Meanline spreadsheet*)
  ρigvoutletrotorinletstaticdensitymin = 1.607539151;
 ρigvoutletrotorinletstaticdensitymax = 1.755055609;
  alphaρigvoutletrotorinletstaticdensity = 0.977708050558412;
  betaρigvoutletrotorinletstaticdensity = 1.02657076522204;
 ρigvoutletrotorinletstaticdensity = InverseBetaRegularized[RandomNumber1,
      alphaρigvoutletrotorinletstaticdensity, betaρigvoutletrotorinletstaticdensity] *
     (ρigvoutletrotorinletstaticdensitymax - ρigvoutletrotorinletstaticdensitymin) +
    ρigvoutletrotorinletstaticdensitymin;


  (*Static Density at Rotor Outlet/Stator Inlet, comes from Meanline spreadsheet*)
  ρrotoroutletstatorinletstaticdensitymin = 1.623776921;
 ρrotoroutletstatorinletstaticdensitymax = 1.772783444;
```

```
alphaρrotoroutletstatorinletstaticdensity = 0.977708011151989;
betaρρrotoroutletstatorinletstaticdensity = 1.02657077653324;
ρrotoroutletstatorinletstaticdensity =
  InverseBetaRegularized[RandomNumber1, alphaρrotoroutletstatorinletstaticdensity,
     betaρρrotoroutletstatorinletstaticdensity] *
    (ρrotoroutletstatorinletstaticdensitymax - ρrotoroutletstatorinletstaticdensitymin) +
   ρrotoroutletstatorinletstaticdensitymin;


(*Meridional Velocity at Fan Inlet, comes from Meanline spreadsheet*)
Cmfaninletmin = 155.0625148;
Cmfaninletmax = 169.291887;
alphaCmfaninlet = 0.999409608906218;
betaCmfaninlet = 1.00406551055043;
Cmfaninlet = InverseBetaRegularized[RandomNumber1, alphaCmfaninlet, betaCmfaninlet] *
    (Cmfaninletmax - Cmfaninletmin) + Cmfaninletmin;
Cmfanoutletstatorinlet = Cmfaninlet * 0.99;
Cmstatoroutletigvinlet = Cmfanoutletstatorinlet * 0.99;
Cmigvoutletrotorinlet = Cmstatoroutletigvinlet * 0.99;
Cmrotoroutletstatorinlet = Cmigvoutletrotorinlet * 0.99;

(*Relative Whirl Velocity at station 1, comes from Meanline spreadsheet*)
We1 = 334.9600142;

(*Relative Whirl Velocity at station 2, comes from Meanline spreadsheet*)
  We2min = 227.9652226;
We2max = 243.5129357;
alphaWe2 = 0.753622390738536;
betaWe2 = 1.01229376849369;
We2 = InverseBetaRegularized[RandomNumber1, alphaWe2, betaWe2] * (We2max - We2min) + We2min;

(*Absolute Whirl Velocity at station 2, comes from Meanline spreadsheet*)
Ce2min = 109.1501745;
Ce2max = 124.6978875;
alphaCe2 = 1.01229368073359;
betaCe2 = 0.753622298219386;
Ce2 = InverseBetaRegularized[RandomNumber1, alphaCe2, betaCe2] * (Ce2max - Ce2min) + Ce2min;

(*Absolute Whirl Velocity at station 3, comes from Meanline spreadsheet*)
Ce3min = 98.65350505;
Ce3max = 116.4677493;
alphaCe3 = 1.02509140073845;
betaCe3 = 0.764970309874072;
Ce3 = InverseBetaRegularized[RandomNumber1, alphaCe3, betaCe3] * (Ce3max - Ce3min) + Ce3min;

(*Absolute Whirl Velocity at station 4, comes from Meanline spreadsheet*)
Ce4min = 86.78654706;
Ce4max = 107.0619107;
alphaCe4 = 1.03423281069255;
betaCe4 = 0.771380676593412;
Ce4 = InverseBetaRegularized[RandomNumber1, alphaCe4, betaCe4] * (Ce4max - Ce4min) + Ce4min;
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)

```
10 |

(*Relative Whirl Velocity at station 4, comes from Meanline spreadsheet*)
We4min = 348.129753;
We4max = 368.4051167;
alphaWe4 = 0.771380636369278;
betaWe4 = 1.03423278242668;
We4 = InverseBetaRegularized[RandomNumber1, alphaWe4, betaWe4] * (We4max - We4min) + We4min;

(*Relative Whirl Velocity at station 5, comes from Meanline spreadsheet*)
We5min = 290.8826549;
We5max = 310.9781211;
alphaWe5 = 0.755498508937843;
betaWe5 = 1.03051523614819;
We5 = InverseBetaRegularized[RandomNumber1, alphaWe5, betaWe5] * (We5max - We5min) + We5min;


InitFanLocAngleWRTParticle = (0.5 - RandomNumber1) * 2 * Pi / nfanblades;
Print["Random Angular Location of Particle at Fan Face (rads): ", InitFanLocAngleWRTParticle];
InitRotorLocAngleWRTParticle = (0.5 - RandomNumber2) * 2 * Pi / nrotorblades;
Print["Random Angular Location of Particle at 1st HP Compressor Face (rads): ",
 InitRotorLocAngleWRTParticle];

nimpactpressuresiderotor = 0;
nimpactsuctionsiderotor = 0;
nnoimpactrotor = 0;
XYLocationofImpactonPressureSiderotor = {};
XYLocationofImpactonSuctionSiderotor = {};
XXPercentChordLocationPressureSiderotor = {};
XXPercentChordLocationSuctionSiderotor = {};
VSuctionSideAppended = {};
VPressureSideAppended = {};
nimpactshaftAppended = 0;
nexitedviaByPassAppended = 0;
RimpactAppendedSuctionSiderotor = {};
RimpactAppendedPressureSiderotor = {};


(* Simulation time increment *)
TimeIncrement = 0.00001;

Nruns = 1; (*number of runs*)
Print["Nrunscount: ", NrunsCount];

Do[Nfanstrikescount = 0;
 Nstatorstrikescount = 0;
 Nigvstrikescount = 0;
 Nrotorstrikescount = 0;
 Nshaftstrikescount = 0;
 Nbypassexitcount = 0;
 Ntaperedfanhousingstrikescount = 0;
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)


```
(*Equations of Motion for Particle in Engine Flow*)
eqns = {0.5 * If[u[t] ≤ 0, ρfaninletstaticdensity,
      If[u[t] ≤ Cfantip * Cos[λtip * P1 / 180], ρfanoutletstatorinletstaticdensity,
       If[Cfantip * Cos[λtip * P1 / 180] < u[t] ≤ (1.2 * Cfantip * Cos[λtip * P1 / 180] +
            Cstatorrms * Cos[λstator * P1 / 180]), ρstatoroutletigvinletstaticdensity,
        If[(1.2 * Cfantip * Cos[λtip * P1 / 180] + Cstatorrms * Cos[λstator * P1 / 180]) <
          u[t] ≤ (1.2 * Cfantip * Cos[λtip * P1 / 180] + 1.2 * Cstatorrms * Cos[
              λstator * P1 / 180] + Cigvrms * Cos[λigv * P1 / 180]),
         ρigvoutletrotorinletstaticdensity, ρrotoroutletstatorinletstaticdensity]]]] *
     (((r'[t])^2 + (r[t] * θ'[t])^2)^0.5) * A0 * (1 - Cdsphere) * r'[t] + m * g * Cos[θ[t]] ==
   -m * (r[t] * (θ'[t])^2 + r''[t]), 0.5 * If[u[t] ≤ 0, ρfaninletstaticdensity,
      If[u[t] ≤ Cfantip * Cos[λtip * P1 / 180], ρfanoutletstatorinletstaticdensity,
       If[Cfantip * Cos[λtip * P1 / 180] < u[t] ≤ (1.2 * Cfantip * Cos[λtip * P1 / 180] +
            Cstatorrms * Cos[λstator * P1 / 180]), ρstatoroutletigvinletstaticdensity,
        If[(1.2 * Cfantip * Cos[λtip * P1 / 180] + Cstatorrms * Cos[λstator * P1 / 180]) < u[t] ≤
          (1.2 * Cfantip * Cos[λtip * P1 / 180] + 1.2 * Cstatorrms * Cos[λstator * P1 / 180] +
            Cigvrms * Cos[λigv * P1 / 180]), ρigvoutletrotorinletstaticdensity,
         ρrotoroutletstatorinletstaticdensity]]] * If[u[t] ≤ Cfantip * Cos[λtip * P1 / 180],
      (r[t] / Rfanrms * (Rfanrms * ωfan - (Rfanrms^3) / (r[t]^2) * ωfan + (Rfanrms^2) / (r[t]^2) *
            (((We2 - r[t] * ωfan) / (Cfantip * Cos[λtip * P1 / 180])) * u[t] + r[t] * ωfan)))^2,
      If[Cfantip * Cos[λtip * P1 / 180] < u[t] ≤ (1.2 * Cfantip * Cos[λtip * P1 / 180] +
          Cstatorrms * Cos[λstator * P1 / 180]),
       -(Rstatorrms / r[t] * ((Ce3 - Ce2) / (0.2 * Cfantip * Cos[λtip * P1 / 180] + Cstatorrms *
                Cos[λstator * P1 / 180]) * u[t] + Ce2 + ((Ce3 - Ce2) * Cfantip * Cos[λtip * P1 / 180]) /
              (0.2 * Cfantip * Cos[λtip * P1 / 180] + Cstatorrms * Cos[λstator * P1 / 180])))^2,
       If[(1.2 * Cfantip * Cos[λtip * P1 / 180] + Cstatorrms * Cos[λstator * P1 / 180]) < u[t] ≤
         (1.2 * Cfantip * Cos[λtip * P1 / 180] + 1.2 * Cstatorrms * Cos[λstator * P1 / 180] +
           Cigvrms * Cos[λigv * P1 / 180]), -(Rigvrms / r[t] * ((Ce4 - Ce3) /
              (0.2 * Cstatorrms * Cos[λstator * P1 / 180] + Cigvrms * Cos[λigv * P1 / 180])) * u[t] + Ce3 +
            ((Ce3 - Ce4) * (1.2 * Cfantip * Cos[λtip * P1 / 180] + Cstatorrms * Cos[λstator * P1 / 180])) /
             (0.2 * Cstatorrms * Cos[λstator * P1 / 180] + Cigvrms * Cos[λigv * P1 / 180])))^2,
        (r[t] / Rrotorrms * (Rrotorrms * ωrotor - (Rrotorrms^3) / (r[t]^2) * ωrotor +
            (Rrotorrms^2) / (r[t]^2) * ((We5 - We4) / (0.2 * Cigvrms * Cos[λigv * P1 / 180] +
                Cistrotortip * Cos[λistrotortip * P1 / 180]) * u[t] + We4 +
              ((We4 - We5) * (1.2 * Cfantip * Cos[λtip * P1 / 180] + 1.2 * Cstatorrms *
                  Cos[λstator * P1 / 180] + Cigvrms * Cos[λigv * P1 / 180])) / (0.2 * Cigvrms *
                Cos[λigv * P1 / 180] + Cistrotortip * Cos[λistrotortip * P1 / 180])))^2]]] *
     A0 * (1 - Cdsphere) - m * g * Sin[θ[t]] == -m * r[t] * θ''[t], 0.5 * If[u[t] ≤ 0,
     ρfaninletstaticdensity,
     If[u[t] ≤ Cfantip * Cos[λtip * P1 / 180],
      ρfanoutletstatorinletstaticdensity,
      If[Cfantip * Cos[λtip * P1 / 180] < u[t] ≤ (1.2 * Cfantip * Cos[λtip * P1 / 180] +
          Cstatorrms * Cos[λstator * P1 / 180]), ρstatoroutletigvinletstaticdensity,
       If[(1.2 * Cfantip * Cos[λtip * P1 / 180] + Cstatorrms * Cos[λstator * P1 / 180]) <
         u[t] ≤ (1.2 * Cfantip * Cos[λtip * P1 / 180] +
           1.2 * Cstatorrms * Cos[λstator * P1 / 180] + Cigvrms * Cos[λigv * P1 / 180]),
        ρigvoutletrotorinletstaticdensity, ρrotoroutletstatorinletstaticdensity]]] *
    If[u[t] ≤ 0, Cmfaninlet^2, If[u[t] ≤ Cfantip * Cos[λtip * P1 / 180],
      Cmfanoutletstatorinlet^2, If[Cfantip * Cos[λtip * P1 / 180] < u[t] ≤
        (1.2 * Cfantip * Cos[λtip * P1 / 180] + Cstatorrms * Cos[λstator * P1 / 180]),
       Cmstatoroutletigvinlet^2, If[(1.2 * Cfantip * Cos[λtip * P1 / 180] +
```

```
12 |
          Cstatorrms * Cos[λstator * Pi / 180]) < u[t] ≤ (1.2 * Cfantip * Cos[λtip * Pi / 180] +
          1.2 * Cstatorrms * Cos[λstator * Pi / 180] + Cigvrms * Cos[λigv * Pi / 180]),
        Cmigvoutletrotorinlet^2, Cmrotoroutletstatorinlet^2]]]] * A0 *
    (1 - Cdsphere) == m * (u''[t]), r[0] == RinitialParticle, r'[
    0] == 0, θ[
    0] == 0, θ'[
    0] == 0.00 *
    ωfan, u[
    0] == 0, u'[
    0] == Vinlet};
(*Numerical Integration Scheme for Particle Motion*)
soln =
 NDSolve[eqns, {r, θ, u}, {t, 0, 0.01}, Method → "ExplicitEuler", "StartingStepSize" → 1 / 10000];

(* Plots only radial particle motion from front of inlet *)
Print[Plot[u[t] /. soln, {t, 0, 0.005},
  PlotTheme → {"OpenMarkersThick", "LargeLabels"}, PlotLegends → u]];
Print[Plot[{Evaluate[{((r'[t] /. soln)^2 + ((r[t] /. soln) * (θ'[t] /. soln))^2)^0.5}],
   r[t] * θ'[t] /. soln, u'[t] /. soln}, {t, 0, 0.005},
  PlotTheme → {"OpenMarkersThick", "LargeLabels"}, PlotLegends →
   {V (Tangential) of Particle Near Fan Blade, R * θ' of Particle Near Fan Blade, u' of Particle}]];
Print[ParametricPlot[{{-(r[t] /. soln) * Sin[θ[t] /. soln], -(r[t] /. soln) * Cos[θ[t] /. soln]},
   {-(r[t] /. soln) * Sin[ωfan * t + InitFanLocAngleWRTParticle],
    -(r[t] /. soln) * Cos[ωfan * t + InitFanLocAngleWRTParticle]},
   {-(r[t] /. soln) * Sin[ωfan * t + Evaluate[2 * Pi / nfanblades] + InitFanLocAngleWRTParticle],
    -(r[t] /. soln) * Cos[ωfan * t + Evaluate[2 * Pi / nfanblades] + InitFanLocAngleWRTParticle]},
   {-Rshaft * Sin[Theta], -Rshaft * Cos[Theta]},
   {-(r[t] /. soln) * Sin[ωfan * t - Evaluate[2 * Pi / nfanblades] - InitFanLocAngleWRTParticle],
    -(r[t] /. soln) * Cos[ωfan * t - Evaluate[2 * Pi / nfanblades] - InitFanLocAngleWRTParticle]}},
  {t, 0, 0.005}, {Theta, 0, Evaluate[2 * Pi]}, PlotTheme → {"OpenMarkersThick", "LargeLabels"},
  PlotLegends → {ParticleDisplacement, FanDisplacementatRparticle,
    NextFanDisplacementatRparticle, Shaft, PreviousFanDisplacementatRparticle}]];

(*Empty Appendable Variables*)
XXfandisplacement = {};
YYfandisplacement = {};
ZZfandisplacement = {};
XXrotordisplacement = {};
YYrotordisplacement = {};
ZZrotordisplacement = {};
YYZZrotordisplacement = {};
XXnextfandisplacement = {};
YYnextfandisplacement = {};
ZZnextfandisplacement = {};
XXpreviousfandisplacement = {};
YYpreviousfandisplacement = {};
ZZpreviousfandisplacement = {};
XXparticledisplacement = {};
YYparticledisplacement = {};
ZZparticledisplacement = {};
XXimpactlocation = {};
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)


```
YYimpactlocation = {};
ZZimpactlocation = {};
XXrotorimpactlocation = {};
YYrotorimpactlocation = {};
ZZrotorimpactlocation = {};
XXROTORimpactlocation = {};
YYROTORimpactlocation = {};
ZZROTORimpactlocation = {};
XXstatorimpactlocation = {};
YYstatorimpactlocation = {};
ZZstatorimpactlocation = {};
XXstatorlocation = {};
YYstatorlocation = {};
ZZstatorlocation = {};
XXigvimpactlocation = {};
YYigvimpactlocation = {};
ZZigvimpactlocation = {};
XXigvlocation = {};
YYigvlocation = {};
ZZigvlocation = {};
DistancetoImpact = {};
DistanceAtStatorImpact = {};
DistanceAtigvImpact = {};
XXnearestnodetoparticle = {};
YYnearestnodetoparticle = {};
ClosestNode = 1;
XXClosestnodetoparticle = 0;
YYClosestnodetoparticle = 0;
SurfaceCounter = 0;
Axialσ = 0;
VSuctionSideAppendedTotal = {};
VPressureSideAppendedTotal = {};

TimeatStartofRotor =
 T /. Extract[FindMinimum[{(u[T] - (1.2 * Cfantip * Cos[λtip * Pi / 180] + 1.2 * Cstatorrms *
             Cos[λstator * Pi / 180] + Cigvrms * Cos[λigv * Pi / 180])) /. soln)^2, {T, 0.001}], {2}];
TimeBetweenParticlevsBladeatStartofRotor = 0;
Print["TimeatStartofRotor: ", TimeatStartofRotor];

(* Particle displacement routine *)
(* Turbomachinery Displacement/Location *)
(* Impact Check Routines - for Fan, Stator and IGV *)

Do[
 If[(Abs[Extract[r[t] /. soln, 1] -
        Rshaft / 2 * (1 + Extract[u[t] /. soln, 1] / (Cfantip * Cos[λtip * Pi / 180]))] ≤
      rfod * Cos[ArcTan[(Rshaft / 2) / (Cfantip * Cos[λtip * Pi / 180])]] &&
     Extract[u[t] /. soln, 1] ≤ Cfantip * Cos[λtip * Pi / 180]) ||
   (Abs[Extract[r[t] /. soln, 1] - Rshaft] ≤ rfod &&
     Extract[u[t] /. soln, 1] > Cfantip * Cos[λtip * Pi / 180]),
  Print[Style["Particle Hit Shaft", FontSize → 48, Red, Bold]];
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)

```
14 |
Print[Style["Time of Shaft Impact: ", FontSize → 48, Red, Bold], t];
Nshaftstrikescount = 1;
Goto[end],
If[(Abs[Extract[r[t] /. soln, 1] - ((Dfanoutlet - Dinlet) *
          Extract[u[t] /. soln, 1] / (2 * Cfantip * Cos[λtip * Pi / 180]) + Dinlet / 2)] ≤
     rfod * Cos[ArcTan[(Dfanoutlet / 2 - Dinlet / 2) / (Cfantip * Cos[λtip * Pi / 180])]] &&
    Extract[u[t] /. soln, 1] ≤ Cfantip * Cos[λtip * Pi / 180]) ||
  (Abs[Extract[r[t] /. soln, 1] - Dfanoutlet / 2] ≤ rfod && 1.2 * Cfantip * Cos[λtip * Pi / 180] +
      Cstatorrms * Cos[λstator * Pi / 180] ≥ Extract[u[t] /. soln, 1] > Cfantip * Cos[λtip * Pi / 180]),
 Print[Style["Particle Struck Engine Housing", FontSize → 48, Red, Bold]];
 Print[Style["Time of Engine Housing Strike: ", FontSize → 48, Red, Bold],
  Style[t, FontSize → 48, Red, Bold]];
 NtaperedfanhousingstrikeScount = 1;
 Goto[end], If[Dstatoroutlet / 2 > Extract[r[t] /. soln, 1] > Digvoutlet / 2 && Extract[u[t] /. soln,
     1] ≥ (1.2 * Cfantip * Cos[λtip * Pi / 180] + Cstatorrms * Cos[λstator * Pi / 180]),
  Print[Style["Particle Exited Engine Via Bypass", FontSize → 48, Red, Bold]];
  Print[Style["Time of ByPass Exit: ", FontSize → 48, Red, Bold],
   Style[t, FontSize → 48, Red, Bold]];
  Nbypassexitcount = 1;
  Goto[end],
  Do[
   Do[
    DistanceAtImpact =
     ((Extract[u[t] /. soln, 1] - ((Xairfoil[[1]] * Cos[TwistTiptoRoot * (Extract[r[t] /. soln, 1] -
                      Rfanrms) / (Dinlet / 2 - Dshaft / 2)] - Yairfoil[[1]] * Sin[TwistTiptoRoot *
                    (Extract[r[t] /. soln, 1] - Rfanrms) / (Dinlet / 2 - Dshaft / 2)]) *
                ((((Extract[r[t] /. soln, 1] - Rfanrms) / HtoCrmsRatio + Crms) / Crms)))^2 +
           (-(Extract[r[t] /. soln, 1]) * Extract[Sin[θ[t] /. soln], 1] -
              ((Xairfoil[[1]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rfanrms) /
                       (Dinlet / 2 - Dshaft / 2)] + Yairfoil[[1]] * Cos[TwistTiptoRoot *
                      (Extract[r[t] /. soln, 1] - Rfanrms) / (Dinlet / 2 - Dshaft / 2)]) *
                  ((((Extract[r[t] /. soln, 1] - Rfanrms) / HtoCrmsRatio + Crms) / Crms) *
                  Cos[ωfan * (t) + InitFanLocAngleWRTParticle + (2 * Pi / nfanblades) * (j - 1)] -
                 (Extract[r[t] /. soln, 1]) * Sin[ωfan * (t) + InitFanLocAngleWRTParticle +
                    (2 * Pi / nfanblades) * (j - 1)])))^2 +
           (-(Extract[r[t] /. soln, 1]) * Extract[Cos[θ[t] /. soln], 1] -
              ((-(Xairfoil[[1]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rfanrms) /
                       (Dinlet / 2 - Dshaft / 2)] + Yairfoil[[1]] * Cos[TwistTiptoRoot *
                      (Extract[r[t] /. soln, 1] - Rfanrms) / (Dinlet / 2 - Dshaft / 2)]) *
                  ((((Extract[r[t] /. soln, 1] - Rfanrms) / HtoCrmsRatio + Crms) / Crms) *
                  Sin[ωfan * (t) + InitFanLocAngleWRTParticle + (2 * Pi / nfanblades) * (j - 1)] -
                 (Extract[r[t] /. soln, 1]) * Cos[ωfan * (t) + InitFanLocAngleWRTParticle +
                    (2 * Pi / nfanblades) * (j - 1)])))^2)^0.5;
    If[DistanceAtImpact ≤ rfod, XXimpactlocation = AppendTo[XXimpactlocation,
       Extract[u[t] /. soln, 1]];
     YYimpactlocation = AppendTo[YYimpactlocation, -(Extract[r[t] /. soln, 1]) *
        Extract[Sin[θ[t] /. soln], 1]];
     ZZimpactlocation = AppendTo[ZZimpactlocation, -(Extract[r[t] /. soln, 1]) *
        Extract[Cos[θ[t] /. soln], 1]];
     Do[
      XXfandisplacement = AppendTo[XXfandisplacement, (Xairfoil[[k]] * Cos[TwistTiptoRoot *
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)


```
            (Extract[r[t] /. soln, 1] - Rfanrms) / (Dinlet / 2 - Dshaft / 2)] - Yairfoil[[k]] *
          Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rfanrms) / (Dinlet / 2 - Dshaft / 2)]) *
        (((Extract[r[t] /. soln, 1] - Rfanrms) / HtoCrmsRatio + Crms) / Crms)];
    YYfandisplacement = AppendTo[YYfandisplacement, (Xairfoil[[k]] * Sin[TwistTiptoRoot *
          (Extract[r[t] /. soln, 1] - Rfanrms) / (Dinlet / 2 - Dshaft / 2)] + Yairfoil[[k]] *
         Cos[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rfanrms) / (Dinlet / 2 - Dshaft / 2)]) *
        (((Extract[r[t] /. soln, 1] - Rfanrms) / HtoCrmsRatio + Crms) / Crms) * Cos[ωfan * (t) +
          InitFanLocAngleWRTParticle + (2 * Pi / nfanblades) * (j - 1)] - (Extract[r[t] /. soln,
          1]) * Sin[ωfan * (t) + InitFanLocAngleWRTParticle + (2 * Pi / nfanblades) * (j - 1)]];
    ZZfandisplacement = AppendTo[ZZfandisplacement,
      ((- (Xairfoil[[k]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rfanrms) /
                (Dinlet / 2 - Dshaft / 2)] + Yairfoil[[k]] * Cos[TwistTiptoRoot *
              (Extract[r[t] /. soln, 1] - Rfanrms) / (Dinlet / 2 - Dshaft / 2)]) *
          (((Extract[r[t] /. soln, 1] - Rfanrms) / HtoCrmsRatio + Crms) / Crms) *
          Sin[ωfan * (t) + InitFanLocAngleWRTParticle +
            (2 * Pi / nfanblades) * (j - 1)] - (Extract[r[t] /. soln, 1]) *
          Cos[ωfan * (t) + InitFanLocAngleWRTParticle + (2 * Pi / nfanblades) * (j - 1)]))],
    {k,
     1,
     51,
     1}];
   Print[Style["Particle Struck Fan", FontSize → 48, Red, Bold]];
   Nfanstrikescount = 1;
   Goto[end]],
  {j, 1, nfanblades}];
 Do[
  DistanceAtStatorImpact =
   ((Extract[u[t] /. soln, 1] - (1.2 * Cfantip * Cos[λtip * Pi / 180] + Xstatorairfoil[[1]]))^2 +
      (- (Extract[r[t] /. soln, 1]) * Extract[Sin[θ[t] /. soln], 1] -
         (Ystatorairfoil[[1]] * Cos[(2 * Pi / nfanblades) * (j - 1) + StatorBladeOffset] -
           (Extract[r[t] /. soln, 1]) * Sin[(2 * Pi / nfanblades) * (j - 1) + StatorBladeOffset]))^
       2 + (- (Extract[r[t] /. soln, 1]) * Extract[Cos[θ[t] /. soln], 1] - (-Ystatorairfoil[[1]] *
           Sin[(2 * Pi / nfanblades) * (j - 1) + StatorBladeOffset] - (Extract[r[t] /. soln, 1]) *
           Cos[(2 * Pi / nfanblades) * (j - 1) + StatorBladeOffset]))^2)^0.5;
  If[DistanceAtStatorImpact <= rfod, XXstatorimpactlocation =
    AppendTo[XXstatorimpactlocation, Extract[u[t] /. soln, 1]];
   YYstatorimpactlocation = AppendTo[YYstatorimpactlocation,
     - (Extract[r[t] /. soln, 1]) * Extract[Sin[θ[t] /. soln], 1]];
   ZZstatorimpactlocation = AppendTo[ZZstatorimpactlocation,
     - (Extract[r[t] /. soln, 1]) * Extract[Cos[θ[t] /. soln], 1]];
   Print[Style["Particle Struck Stator", FontSize → 48, Red, Bold]];
   Nstatorstrikescount = 1;
   Goto[end]], {j, 1, nfanblades}];
 Do[
  DistanceAtigvImpact = ((Extract[u[t] /. soln, 1] - (1.2 * Cfantip * Cos[λtip * Pi / 180] +
          1.2 * Cstatorrms * Cos[λstator * Pi / 180] + Xigvairfoil[[1]]))^2 +
      (- (Extract[r[t] /. soln, 1]) * Extract[Sin[θ[t] /. soln], 1] - (Yigvairfoil[[1]] *
           Cos[(2 * Pi / nrotorblades) * (j - 1) + IGVBladeOffset] - (Extract[r[t] /. soln, 1]) *
           Sin[(2 * Pi / nrotorblades) * (j - 1) + IGVBladeOffset]))^2 +
      (- (Extract[r[t] /. soln, 1]) * Extract[Cos[θ[t] /. soln], 1] - (-Yigvairfoil[[1]] *
           Sin[(2 * Pi / nrotorblades) * (j - 1) + IGVBladeOffset] - (Extract[r[t] /. soln, 1]) *
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)

```
16 |                    Cos[(2 * Pi / nrotorblades) * (j - 1) + IGVBladeOffset]))^2)^0.5;
        If[DistanceAtIgvImpact ≤ rfod, XXIgvimpactlocation = AppendTo[
           XXIgvimpactlocation, Extract[u[t] /. soln, 1]];
         YYIgvimpactlocation = AppendTo[YYIgvimpactlocation,
           -(Extract[r[t] /. soln, 1]) * Extract[Sin[e[t] /. soln], 1]];
         ZZIgvimpactlocation = AppendTo[ZZIgvimpactlocation,
           -(Extract[r[t] /. soln, 1]) * Extract[Cos[e[t] /. soln], 1]];
         Print[Style["Particle Struck IGV", FontSize → 48, Red, Bold]];
         NigvstrikesCount = 1;
         Goto[end]], {j, 1, nrotorblades}], {i, 1, 51, 1}];
      XXparticledisplacement = AppendTo[XXparticledisplacement, u[t] /. soln];
      YYparticledisplacement = AppendTo[YYparticledisplacement, -(r[t] /. soln) * Sin[e[t] /. soln]];
      ZZparticledisplacement =
       AppendTo[ZZparticledisplacement, -(r[t] /. soln) * Cos[e[t] /. soln]]]]
   , {t, 0, 0.0050, TimeIncrement}];


(*Impact Check Routines - for 1st Rotor set also extract location of impact*)
Do[SurfaceLocator = 0;
 Do[
  Do[
   DistanceAtRotorImpact =
    ((Extract[u[t] /. soln, 1] - (1.2 * Cfantip * Cos[λtip * Pi / 180] + 1.2 * Cstatorrms * Cos[λstator * Pi /
                180] + 1.2 * Cigvrms * Cos[λigv * Pi / 180] + (Xrotorairfoil[[i]] * Cos[TwistTiptoRoot *
                  (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)] -
               Yrotorairfoil[[i]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                  (Digvoutlet / 2 - Dshaft / 2)]) * (((Extract[r[t] /. soln, 1] - Rrotorrms) /
                HtoCrmsRatioIstrotor + Cistrotorrms) / Cistrotorrms)))^
        2 + (-(Extract[r[t] /. soln, 1]) * Extract[Sin[e[t] /. soln], 1] -
          ((Xrotorairfoil[[i]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                  (Digvoutlet / 2 - Dshaft / 2)] + Yrotorairfoil[[i]] * Cos[TwistTiptoRoot *
                (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) *
             (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioIstrotor + Cistrotorrms) /
              Cistrotorrms) * Cos[ωrotor * (t + TimeBetweenParticlevsBladeatStartofRotor) +
              InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)] - (Extract[
              r[t] /. soln, 1]) * Sin[ωrotor * (t + TimeBetweenParticlevsBladeatStartofRotor) +
              InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]))^2 +
        (-(Extract[r[t] /. soln, 1]) * Extract[Cos[e[t] /. soln], 1] -
          ((-(Xrotorairfoil[[i]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                  (Digvoutlet / 2 - Dshaft / 2)] + Yrotorairfoil[[i]] * Cos[TwistTiptoRoot *
                (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) *
             (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioIstrotor + Cistrotorrms) /
              Cistrotorrms) * Sin[ωrotor * (t + TimeBetweenParticlevsBladeatStartofRotor) +
              InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)] - (Extract[
              r[t] /. soln, 1]) * Cos[ωrotor * (t + TimeBetweenParticlevsBladeatStartofRotor) +
              InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)])))^2)^0.5;
    If[DistanceAtRotorImpact ≤ rfod, XXROTORimpactlocation = AppendTo[
       XXROTORimpactlocation, Extract[u[t] /. soln, 1]];
     YYROTORimpactlocation = AppendTo[YYROTORimpactlocation,
       -(Extract[r[t] /. soln, 1]) * Extract[Sin[e[t] /. soln], 1]];
     YYPlaneImpactLocation = YYROTORimpactlocation /
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)


```
    (Cos[ωrotor * (t + TimeBetweenParticlesvsBladeatStartofRotor) +
        InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]);
ZZROTORimpactlocation = AppendTo[ZZROTORimpactlocation,
  -(Extract[r[t] /. soln, 1]) * Extract[Cos[θ[t] /. soln], 1]];
Do[dydu = -(Extract[r[t] /. soln, 1] * Extract[Cos[θ[t] /. soln], 1] * Extract[θ'[t] /. soln, 1] +
      Extract[Sin[θ[t] /. soln], 1] * Extract[r'[t] /. soln, 1]) / (Extract[u'[t] /. soln, 1]);
 If[((Extract[u[t] /. soln, 1] - (1.2 * Cfantip * Cos[λtip * Pi / 180] +
           1.2 * Cstatorrms * Cos[λstator * Pi / 180] + 1.2 * Cigvrms * Cos[λigv * Pi / 180] +
           (Xrotorairfoil[[n]] * Cos[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                 (Digvoutlet / 2 - Dshaft / 2)] - Yrotorairfoil[[n]] * Sin[TwistTiptoRoot *
                (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) *
            (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioIstrotor + Cistrotorrms) /
              Cistrotorrms)))^2 + (-(Extract[r[t] /. soln, 1]) * Extract[Sin[θ[t] /. soln], 1] -
          ((Xrotorairfoil[[n]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                 (Digvoutlet / 2 - Dshaft / 2)] + Yrotorairfoil[[n]] * Cos[TwistTiptoRoot *
                (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) *
            (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioIstrotor + Cistrotorrms) /
              Cistrotorrms) * Cos[ωrotor * (t + TimeBetweenParticlesvsBladeatStartofRotor) +
              InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)] - (Extract[
              r[t] /. soln, 1]) * Sin[ωrotor * (t + TimeBetweenParticlesvsBladeatStartofRotor) +
              InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]))^2 +
        (-(Extract[r[t] /. soln, 1]) * Extract[Cos[θ[t] /. soln], 1] -
          ((-(Xrotorairfoil[[n]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                 (Digvoutlet / 2 - Dshaft / 2)] + Yrotorairfoil[[n]] * Cos[TwistTiptoRoot *
                (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) *
            (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioIstrotor + Cistrotorrms) /
              Cistrotorrms) * Sin[ωrotor * (t + TimeBetweenParticlesvsBladeatStartofRotor) +
              InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)] - (Extract[
              r[t] /. soln, 1]) * Cos[ωrotor * (t + TimeBetweenParticlesvsBladeatStartofRotor) +
              InitRotorLocAngleWRTParticle +
              (2 * Pi / nrotorblades) * (j - 1)]))^2)^0.5 ≤ ClosestNode,

  Nimpact = n;

  Rimpact = Extract[r[t] /. soln, 1];
  If[n ≤ 26, SurfaceLocator = "Suction Side", SurfaceLocator = "Pressure Side"];
  If[n == 26, LocalThicknessatImpact = 0, LocalThicknessatImpact =
    ((((Xrotorairfoil[[n]] * Cos[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                 (Digvoutlet / 2 - Dshaft / 2)] - Yrotorairfoil[[n]] * Sin[TwistTiptoRoot *
                (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) *
            (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioIstrotor + Cistrotorrms) /
              Cistrotorrms)) - ((Xrotorairfoil[[52 - n]] * Cos[TwistTiptoRoot *
                (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)] -
              Yrotorairfoil[[52 - n]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] -
                    Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) * (((Extract[r[t] /. soln, 1] -
                  Rrotorrms) / HtoCrmsRatioIstrotor + Cistrotorrms) / Cistrotorrms)))^2 +
        ((((Xrotorairfoil[[n]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] -
                      Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)] + Yrotorairfoil[[n]] *
                  Cos[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                    (Digvoutlet / 2 - Dshaft / 2)]) * (((Extract[r[t] /. soln, 1] - Rrotorrms) /
                  HtoCrmsRatioIstrotor + Cistrotorrms) / Cistrotorrms) *
```

```
18 |
                    Cos[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                      InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) *
                       (j - 1)] - (Extract[r[t] /. soln, 1]) *
                    Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                      InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) *
                       (j - 1)] + (Extract[r[t] /. soln, 1]) *
                    Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                      InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) /
                  (Cos[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                     InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) - (Extract[
                    r[t] /. soln, 1]) * Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                    InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) -
                  (((XrotorairfoilTip[[52 - n]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] -
                          Rrotorrms) / (DigvoutletTip / 2 - Dshaft / 2)] + YrotorairfoilTip[[52 - n]] *
                       Cos[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                          (DigvoutletTip / 2 - Dshaft / 2)]) * (((Extract[r[t] /. soln, 1] - Rrotorrms) /
                        HtoCrmsRatioTipstrotor + CtipstrotorrmsTip) / CtipstrotorrmsTip) *
                    Cos[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                      InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) *
                       (j - 1)] - (Extract[r[t] /. soln, 1]) *
                    Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                      InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) *
                       (j - 1)] + (Extract[r[t] /. soln, 1]) *
                    Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                      InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) /
                  (Cos[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                     InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) - (Extract[
                    r[t] /. soln, 1]) * Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                    InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)])^2)^0.5];
ClosestNode = ((Extract[u[t] /. soln, 1] - (1.2 * CfantipTip * Cos[λtip * Pi / 180] +
            1.2 * CstatorrmsTip * Cos[λstator * Pi / 180] + 1.2 * CigvrmsTip * Cos[λigv * Pi / 180] +
            (XrotorairfoilTip[[n]] * Cos[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                  (DigvoutletTip / 2 - Dshaft / 2)] - YrotorairfoilTip[[n]] * Sin[TwistTiptoRoot *
                  (Extract[r[t] /. soln, 1] - Rrotorrms) / (DigvoutletTip / 2 - Dshaft / 2)]) *
             (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioTipstrotor + CtipstrotorrmsTip) /
              CtipstrotorrmsTip)))^2 + (- (Extract[r[t] /. soln, 1]) * Extract[Sin[θ[t] /. soln], 1] -
          ((XrotorairfoilTip[[n]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                  (DigvoutletTip / 2 - Dshaft / 2)] + YrotorairfoilTip[[n]] * Cos[TwistTiptoRoot *
                  (Extract[r[t] /. soln, 1] - Rrotorrms) / (DigvoutletTip / 2 - Dshaft / 2)]) *
             (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioTipstrotor + CtipstrotorrmsTip) /
              CtipstrotorrmsTip) * Cos[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
              InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)] - (Extract[
              r[t] /. soln, 1]) * Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
              InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]))^2 +
        (- (Extract[r[t] /. soln, 1]) * Extract[Cos[θ[t] /. soln], 1] -
          ((- (XrotorairfoilTip[[n]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                  (DigvoutletTip / 2 - Dshaft / 2)] + YrotorairfoilTip[[n]] * Cos[TwistTiptoRoot *
                  (Extract[r[t] /. soln, 1] - Rrotorrms) / (DigvoutletTip / 2 - Dshaft / 2)]) *
             (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioTipstrotor + CtipstrotorrmsTip) /
              CtipstrotorrmsTip) * Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
              InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)] - (Extract[
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)

```
                r[t] /. soln, 1]) * Cos[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                 InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]])^2)^0.5;
XXClosestnodetoparticle = (Xrotorairfoil[[n]] * Cos[TwistTiptoRoot * (Extract[r[t] /. soln,
          1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)] - Yrotorairfoil[[n]] * Sin[
      TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) *
   (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioistrotor + Cistrotorrms) /
     Cistrotorrms);
YYClosestnodetoparticle = ((Xrotorairfoil[[n]] * Sin[TwistTiptoRoot *
           (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)] +
        Yrotorairfoil[[n]] * Cos[TwistTiptoRoot * (Extract[r[t] /. soln, 1] -
            Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) *
      (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioistrotor + Cistrotorrms) /
        Cistrotorrms) * Cos[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
        InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)] -
     (Extract[r[t] /. soln, 1]) * Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
        InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)] *
     (Extract[r[t] /. soln, 1]) * Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
        InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) /
   (Cos[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
      InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) -
  (Extract[r[t] /. soln, 1]) * Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
     InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)];
Chord = ((((Xrotorairfoil[[1]] * Cos[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                 (Digvoutlet / 2 - Dshaft / 2)] - Yrotorairfoil[[1]] * Sin[TwistTiptoRoot *
                 (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) *
            (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioistrotor + Cistrotorrms) /
              Cistrotorrms)) - ((Xrotorairfoil[[26]] * Cos[TwistTiptoRoot *
                 (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)] -
              Yrotorairfoil[[26]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                 (Digvoutlet / 2 - Dshaft / 2)]) * (((Extract[r[t] /. soln, 1] - Rrotorrms) /
                HtoCrmsRatioistrotor + Cistrotorrms) / Cistrotorrms)))^2 +
     ((((Xrotorairfoil[[1]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                   (Digvoutlet / 2 - Dshaft / 2)] + Yrotorairfoil[[1]] * Cos[TwistTiptoRoot *
                   (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) *
               (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioistrotor + Cistrotorrms) /
                 Cistrotorrms) * Cos[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                 InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) *
                  (j - 1)] - (Extract[r[t] /. soln, 1]) *
               Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                 InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) *
                  (j - 1)] + (Extract[r[t] /. soln, 1]) *
               Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                 InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) /
             (Cos[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) - (Extract[
               r[t] /. soln, 1]) * Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
               InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) -
           (((Xrotorairfoil[[26]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                   (Digvoutlet / 2 - Dshaft / 2)] + Yrotorairfoil[[26]] * Cos[TwistTiptoRoot *
                   (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) *
               (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioistrotor + Cistrotorrms) /
```

```
20 |
                    Cistrotorrms) * Cos[ωrotor * (t + TimeBetweenParticlevsBladeatStartofRotor) +
                    InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) *
                     (j - 1)] - (Extract[r[t] /. soln, 1]) *
                   Sin[ωrotor * (t + TimeBetweenParticlevsBladeatStartofRotor) +
                    InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) *
                     (j - 1)] + (Extract[r[t] /. soln, 1]) *
                   Sin[ωrotor * (t + TimeBetweenParticlevsBladeatStartofRotor) +
                    InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) /
                 (Cos[ωrotor * (t + TimeBetweenParticlevsBladeatStartofRotor) +
                    InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) - (Extract[
                   r[t] /. soln, 1]) * Sin[ωrotor * (t + TimeBetweenParticlevsBladeatStartofRotor) +
                    InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)])) ^ 2) ^ 0.5;
XpercentChord = ((YYClosestnodetoparticle - ((Xrotorairfoil[[26]] *
                   Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                     (Digvoutlet / 2 - Dshaft / 2)] + Yrotorairfoil[[26]] * Cos[TwistTiptoRoot *
                     (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) *
                  (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioistrotor + Cistrotorrms) /
                    Cistrotorrms) * Cos[ωrotor * (t + TimeBetweenParticlevsBladeatStartofRotor) +
                    InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)] - (Extract[
                   r[t] /. soln, 1]) * Sin[ωrotor * (t + TimeBetweenParticlevsBladeatStartofRotor) +
                    InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)])) *
              Sin[(TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                   (Digvoutlet / 2 - Dshaft / 2) + λistrotortip * Pi / 180)] +
             (XXClosestnodetoparticle - (Xrotorairfoil[[26]] * Cos[TwistTiptoRoot *
                   (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)] -
                  Yrotorairfoil[[26]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                   (Digvoutlet / 2 - Dshaft / 2)]) * (((Extract[r[t] /. soln, 1] - Rrotorrms) /
                   HtoCrmsRatioistrotor + Cistrotorrms) / Cistrotorrms)) *
              Cos[(TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 -
                   Dshaft / 2) + λistrotortip * Pi / 180)]) / Chord;
If[n ≤ 26, If[n == 26, VSuctionSide = ((Extract[u'[t] /. soln, 1]) ^ 2 + (Extract[r[t] /. soln, 1] *
                 Extract[Sin[θ[t] /. soln], 1] * Extract[θ'[t] /. soln, 1] - Extract[Cos[θ[t] /.
                   soln], 1] * Extract[r'[t] /. soln, 1]) ^ 2) ^ 0.5, ThetaImpactSiteSuctionSide =
            ArcTan[(((Xrotorairfoil[[n]] * Cos[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                     (Digvoutlet / 2 - Dshaft / 2)] - Yrotorairfoil[[n]] * Sin[TwistTiptoRoot *
                     (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) *
                   (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioistrotor + Cistrotorrms) /
                    Cistrotorrms)) - ((Xrotorairfoil[[n + 1]] * Cos[TwistTiptoRoot *
                     (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)] -
                    Yrotorairfoil[[n + 1]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] -
                       Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) * (((Extract[r[t] /. soln, 1] -
                       Rrotorrms) / HtoCrmsRatioistrotor + Cistrotorrms) / Cistrotorrms))) /
              ((((Xrotorairfoil[[n]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                     (Digvoutlet / 2 - Dshaft / 2)] + Yrotorairfoil[[n]] * Cos[TwistTiptoRoot *
                     (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) *
                   (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioistrotor + Cistrotorrms) /
                    Cistrotorrms) * Cos[ωrotor * (t + TimeBetweenParticlevsBladeatStartofRotor) +
                    InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) *
                     (j - 1)] - (Extract[r[t] /. soln, 1]) *
                   Sin[ωrotor * (t + TimeBetweenParticlevsBladeatStartofRotor) +
                    InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) *
```

```
                (j - 1)] + (Extract[r[t] /. soln, 1]) *
              Sin[ωrotor * (t + TimeBetweenParticlesVsBladeatStartofRotor) +
                InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) /
            (Cos[ωrotor * (t + TimeBetweenParticlesVsBladeatStartofRotor) +
               InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) - (Extract[
              r[t] /. soln, 1]) * Sin[ωrotor * (t + TimeBetweenParticlesVsBladeatStartofRotor) +
              InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) -
          (((Xrotorairfoil[[n + 1]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] -
                      Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)] + Yrotorairfoil[[n + 1]] *
                  Cos[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                    (Digvoutlet / 2 - Dshaft / 2)]) * (((Extract[r[t] /. soln, 1] - Rrotorrms) /
                  HtoCrmsRatioistrotor + Cistrotorrms) / Cistrotorrms) *
                Cos[ωrotor * (t + TimeBetweenParticlesVsBladeatStartofRotor) +
                  InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) *
                   (j - 1)] - (Extract[r[t] /. soln, 1]) *
                Sin[ωrotor * (t + TimeBetweenParticlesVsBladeatStartofRotor) +
                  InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) *
                   (j - 1)] + (Extract[r[t] /. soln, 1]) *
                Sin[ωrotor * (t + TimeBetweenParticlesVsBladeatStartofRotor) +
                  InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) /
              (Cos[ωrotor * (t + TimeBetweenParticlesVsBladeatStartofRotor) +
                 InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) - (Extract[
                r[t] /. soln, 1]) * Sin[ωrotor * (t + TimeBetweenParticlesVsBladeatStartofRotor) +
                InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]))];
      VSuctionSide = ((Extract[u'[t] /. soln, 1] * Cos[ThetaImpactSiteSuctionSide - dydu])^2 +
          (Extract[r[t] /. soln, 1] * Extract[Sin[θ[t] /. soln], 1] * Extract[θ'[t] /. soln, 1] -
            Extract[Cos[θ[t] /. soln], 1] * Extract[r'[t] /. soln, 1])^2)^0.5],
     If[n == 51, VPressureSide = (((Extract[r[t] /. soln, 1] * ωrotor - (Extract[
                  r[t] /. soln, 1] * Extract[Cos[θ[t] /. soln], 1] * Extract[θ'[t] /. soln, 1] +
                Extract[Sin[θ[t] /. soln], 1] * Extract[r'[t] /. soln, 1]))^2 +
          (Extract[r[t] /. soln, 1] * Extract[Sin[θ[t] /. soln], 1] * Extract[θ'[t] /. soln, 1] -
            Extract[Cos[θ[t] /. soln], 1] * Extract[r'[t] /. soln, 1])^2)^0.5,
      ThetaImpactSitePressureSide = ArcTan[(((Xrotorairfoil[[n + 1]] * Cos[TwistTiptoRoot *
                  (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)] -
              Yrotorairfoil[[n + 1]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] -
                  Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) * (((Extract[r[t] /. soln, 1] -
                  Rrotorrms) / HtoCrmsRatioistrotor + Cistrotorrms) / Cistrotorrms)) -
            ((Xrotorairfoil[[n]] * Cos[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                  (Digvoutlet / 2 - Dshaft / 2)] - Yrotorairfoil[[n]] * Sin[TwistTiptoRoot *
                (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) *
              (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioistrotor + Cistrotorrms) /
                Cistrotorrms))) / ((((Xrotorairfoil[[n + 1]] * Sin[TwistTiptoRoot *
                    (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)] +
                Yrotorairfoil[[n + 1]] * Cos[TwistTiptoRoot * (Extract[r[t] /. soln, 1] -
                      Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) * (((Extract[r[t] /. soln,
                    1] - Rrotorrms) / HtoCrmsRatioistrotor + Cistrotorrms) / Cistrotorrms) *
              Cos[ωrotor * (t + TimeBetweenParticlesVsBladeatStartofRotor) +
                InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) *
                 (j - 1)] - (Extract[r[t] /. soln, 1]) *
              Sin[ωrotor * (t + TimeBetweenParticlesVsBladeatStartofRotor) +
                InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) *
```

```
22 |
                    (j - 1)] + (Extract[r[t] /. soln, 1]) *
                  Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                    InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) /
                (Cos[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                   InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) - (Extract[
                  r[t] /. soln, 1]) * Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                  InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) -
              (((XrotorairfoilTip[[n]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
                      (Digvoutlet / 2 - Dshaft / 2)] + YrotorairfoilTip[[n]] * Cos[TwistTiptoRoot *
                     (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) *
                  (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioistrotor + Cistrotorrms) /
                    Cistrotorrms) * Cos[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                    InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) *
                     (j - 1)] - (Extract[r[t] /. soln, 1]) *
                  Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                    InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) *
                     (j - 1)] + (Extract[r[t] /. soln, 1]) *
                  Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                    InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) /
                (Cos[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                   InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) - (Extract[
                  r[t] /. soln, 1]) * Sin[ωrotor * (t + TimeBetweenParticlesBladeatStartofRotor) +
                  InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]))];
    VPressureSide = (((Extract[r[t] /. soln, 1] * ωrotor - (Extract[r[t] /. soln, 1] * Extract[
                  Cos[θ[t] /. soln], 1] * Extract[θ'[t] /. soln, 1] + Extract[Sin[θ[t] /. soln],
                  1] * Extract[r'[t] /. soln, 1])) * Sin[ThetaImpactSitePressureSide - dydu]) ^ 2 +
            (Extract[r[t] /. soln, 1] * Extract[Sin[θ[t] /. soln], 1] * Extract[θ'[t] /. soln, 1] -
              Extract[Cos[θ[t] /. soln], 1] * Extract[r'[t] /. soln, 1]) ^ 2) ^ 0.5]]
];

AxialσTip = ((1 / (192 (CistrotorrmsTip HtoCrmsRatioistrotor + Extract[r[t] /. soln, 1] - Rrotorrms)^2))
    (3 DistrotoroutletTip^4 + 16 DistrotoroutletTip^3 (CistrotorrmsTip HtoCrmsRatioistrotor - Rrotorrms) +
      24 DistrotoroutletTip^2 (-CistrotorrmsTip HtoCrmsRatioistrotor + Rrotorrms)^2 - 16 Extract[
        r[t] /. soln, 1]^2 (6 CistrotorrmsTip^2 HtoCrmsRatioistrotor^2 + 3 Extract[r[t] /. soln, 1]^2 +
        4 CistrotorrmsTip HtoCrmsRatioistrotor (2 Extract[r[t] /. soln, 1] - 3 Rrotorrms) - 8
         Extract[r[t] /. soln, 1] Rrotorrms + 6 Rrotorrms^2)) ωrotor^2 ρT16A14v),

(n, 1, 51, 1)];

Do[
 XXrotordisplacement = AppendTo[XXrotordisplacement, 1.2 * CfantipTip * Cos[λtip * Pi / 180] +
    1.2 * Cstatorrms * Cos[λstator * Pi / 180] + 1.2 * Cigvrms * Cos[λigv * Pi / 180] +
    (XrotorairfoilTip[[k]] * Cos[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
           (Digvoutlet / 2 - Dshaft / 2)] - YrotorairfoilTip[[k]] * Sin[TwistTiptoRoot *
          (Extract[r[t] /. soln, 1] - Rrotorrms) / (Digvoutlet / 2 - Dshaft / 2)]) *
     (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioistrotor + Cistrotorrms) /
       Cistrotorrms)];
 XXplanerotordisplacement = XXrotordisplacement - (1.2 * CfantipTip * Cos[λtip * Pi / 180] +
     1.2 * Cstatorrms * Cos[λstator * Pi / 180] + 1.2 * Cigvrms * Cos[λigv * Pi / 180]);
 YYrotordisplacement = AppendTo[YYrotordisplacement,
   (XrotorairfoilTip[[k]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)


```
            (DigvoutlOt / 2 - Dshaft / 2)] + YrotorairfoilI[k]] * Cos[TwistTiptoRoot *
           (Extract[r[t] /. soln, 1] - Rrotorrms) / (DigvoutlOt / 2 - Dshaft / 2)]) *
      (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioIstrotor + CIstrotorrms) /
        CIstrotorrms) * Cos[wrotor * (t + TimeBetweenParticlesvsBladeatStartofRotor) +
        InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)] -
    (Extract[r[t] /. soln, 1]) * Sin[wrotor * (t + TimeBetweenParticlesvsBladeatStartofRotor) +
        InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]];
YYplanerotordisplacement = (YYrotordisplacement + (Extract[r[t] /. soln, 1]) *
       Sin[wrotor * (t + TimeBetweenParticlesvsBladeatStartofRotor) +
         InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) /
    (Cos[wrotor * (t + TimeBetweenParticlesvsBladeatStartofRotor) +
       InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) -
   (Extract[r[t] /. soln, 1]) * Sin[wrotor * (t + TimeBetweenParticlesvsBladeatStartofRotor) +
      InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)];
ZZrotordisplacement = AppendTo[ZZrotordisplacement,
   (- (XrotorairfoilI[k]] * Sin[TwistTiptoRoot * (Extract[r[t] /. soln, 1] - Rrotorrms) /
             (DigvoutlOt / 2 - Dshaft / 2)] + YrotorairfoilI[k]] * Cos[TwistTiptoRoot *
           (Extract[r[t] /. soln, 1] - Rrotorrms) / (DigvoutlOt / 2 - Dshaft / 2)]) *
      (((Extract[r[t] /. soln, 1] - Rrotorrms) / HtoCrmsRatioIstrotor + CIstrotorrms) /
        CIstrotorrms) * Sin[wrotor * (t + TimeBetweenParticlesvsBladeatStartofRotor) +
        InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)] -
    (Extract[r[t] /. soln, 1]) * Cos[wrotor * (t + TimeBetweenParticlesvsBladeatStartofRotor) +
        InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)])];
ZZplanerotordisplacement = (ZZrotordisplacement + (Extract[r[t] /. soln, 1]) *
       Cos[wrotor * (t + TimeBetweenParticlesvsBladeatStartofRotor) +
         InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) /
    (Sin[wrotor * (t + TimeBetweenParticlesvsBladeatStartofRotor) +
       InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)]) -
   (Extract[r[t] /. soln, 1]) * Cos[wrotor * (t + TimeBetweenParticlesvsBladeatStartofRotor) +
      InitRotorLocAngleWRTParticle + (2 * Pi / nrotorblades) * (j - 1)],
 {k,
  1,
  51,
  1}];
Print[Style["Particle Struck Rotor", FontSize → 48, Red, Bold]];
NrotorstrikescountI = 1; Goto[end], Rimpact = 0;
XpercentChord = 0;
SurfaceLocator = 0; VSuctionSide = 0;
VPressureSide = 0;
Axialc = 0;
DepthofPenetration = 0;
PenetrationDiam = 0;
ChordatImpact = 0;
BetaCFC = 0;
ImpactEccentricty = 0;
AverageAirfoilThickness = 0;
G1 = 0;
G2 = 0;
G3 = 0;
G4 = 0;
BetaW = 0;
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)

```
24 |
      BetaFFS = 0;
      BetaCompounded = 0;
      LocalThicknessatImpact = 0;
      Nimpact = 0], {j, 1, nrotorblades}], {i, 1, 51, 1}],
   {t, TimeatStartofRotor, 0.0050, TimeIncrement}];

Label[end];


NshaftstrikesTotal = NshaftstrikesTotal + Nshaftstrikescount;
NbypassexitTotal = NbypassexitTotal + Nbypassexitcount;
NtaperedfanhousingstrikesTotal =
 NtaperedfanhousingstrikesTotal + Ntaperedfanhousingstrikescount;
NfanstrikesTotal = NfanstrikesTotal + Nfanstrikescount;
NstatorstrikesTotal = NstatorstrikesTotal + Nstatorstrikescount;
NigvstrikesTotal = NigvstrikesTotal + Nigvstrikescount;
NrotorstrikesTotal = NrotorstrikesTotal + Nrotorstrikescount;


ROTORImpactLocationCoordinates =
 Transpose[{XXROTORimpactlocation, YYROTORimpactlocation, ZZROTORimpactlocation}];

(* Stator & IGV Location *)
Do[Do[Do[XXstatorlocation =
    AppendTo[XXstatorlocation, 1.2 * Cfantip * Cos[λtip * Pi / 180] + Xstatorairfoil[[i]]];
   YYstatorlocation = AppendTo[YYstatorlocation,
     (Ystatorairfoil[[i]] * Cos[(2 * Pi / nfanblades) * (j - 1) + StatorBladeOffset]) -
      (rr) * Sin[(2 * Pi / nfanblades) * (j - 1) + StatorBladeOffset]];
   ZZstatorlocation = AppendTo[ZZstatorlocation,
     (-Ystatorairfoil[[i]] * Sin[(2 * Pi / nfanblades) * (j - 1) + StatorBladeOffset]) -
      (rr) * Cos[(2 * Pi / nfanblades) * (j - 1) + StatorBladeOffset]], {i, 1, 51}],
  {rr, 0.21, Dfanoutlet / 2, StatorAirfoilSpacing}], {j, 1, nfanblades}];

StatorLocationCoordinates = Transpose[{XXstatorlocation, YYstatorlocation, ZZstatorlocation}];
StatorImpactLocationCoordinates =
 Transpose[{XXstatorimpactlocation, YYstatorimpactlocation, ZZstatorimpactlocation}];

Do[Do[Do[XXigvlocation = AppendTo[XXigvlocation,
     1.2 * Cfantip * Cos[λtip * Pi / 180] + 1.2 * Cstatorrms * Cos[λstator * Pi / 180] + Xigvairfoil[[i]]];
   YYigvlocation = AppendTo[YYigvlocation, (Yigvairfoil[[i]] * Cos[(2 * Pi / nrotorblades) * (j - 1) +
         IGVBladeOffset]) - (rr) * Sin[(2 * Pi / nrotorblades) * (j - 1) + IGVBladeOffset]];
   ZZigvlocation = AppendTo[ZZigvlocation, (-Yigvairfoil[[i]] * Sin[(2 * Pi / nrotorblades) * (j - 1) +
         IGVBladeOffset]) - (rr) * Cos[(2 * Pi / nrotorblades) * (j - 1) + IGVBladeOffset]], {i, 1, 51}],
  {rr, 0.21, Digvoutlet / 2, IGVAirfoilSpacing}], {j, 1, nrotorblades}];
igvLocationCoordinates = Transpose[{XXigvlocation, YYigvlocation, ZZigvlocation}];
igvImpactLocationCoordinates =
 Transpose[{XXigvimpactlocation, YYigvimpactlocation, ZZigvimpactlocation}];

Print["Fan XXimpactlocation: ", XXimpactlocation];
Print["Fan YYimpactlocation: ", YYimpactlocation];
Print["Fan ZZimpactlocation: ", ZZimpactlocation];
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)


```
ImpactLocationCoordinates = Transpose[{XXimpactlocation, YYimpactlocation, ZZimpactlocation}];
FanDisplacementCoordinates =
 Transpose[{XXfandisplacement, Flatten[YYfandisplacement], Flatten[ZZfandisplacement]}];
RotorImpactLocationCoordinates = Transpose[
  {XXrotorimpactlocation, YYrotorimpactlocation, ZZrotorimpactlocation}];
RotorDisplacementCoordinates = Transpose[{XXrotordisplacement,
   Flatten[YYrotordisplacement], Flatten[ZZrotordisplacement]}];
NextFanDisplacementCoordinates = Transpose[{XXnextfandisplacement,
   Flatten[YYnextfandisplacement], Flatten[ZZnextfandisplacement]}];
PreviousFanDisplacementCoordinates = Transpose[{XXpreviousfandisplacement,
   Flatten[YYpreviousfandisplacement], Flatten[ZZpreviousfandisplacement]}];
YZfandisplacementcoordinates = Transpose[{Flatten[YYfandisplacement],
   Flatten[ZZfandisplacement]}];
Graphics3D[Point[FanDisplacementCoordinates]];
ParticleDisplacementCoordinates = Transpose[{Flatten[XXparticledisplacement],
   Flatten[YYparticledisplacement], Flatten[ZZparticledisplacement]}];
YZparticledisplacementcoordinates = Transpose[
  {Flatten[YYparticledisplacement], Flatten[ZZparticledisplacement]}];
Print[ListPlot[YZparticledisplacementcoordinates, GridLines → Automatic]];
Print["ROTOR XXimpactlocation: ", XXROTORimpactlocation];
Print["ROTOR YYimpactlocation: ", YYROTORimpactlocation];
Print["ROTOR ZZimpactlocation: ", ZZROTORimpactlocation];

XYrotorbladecoordinatesaroundimpact =
 Transpose[{Flatten[{XXrotordisplacement - (1.2 * Cfantip * Cos[λtip * P1 / 180] + 1.2 * Cstatorrms * Cos[
          λstator * P1 / 180] + 1.2 * Cigvrms * Cos[λigv * P1 / 180]))}], Flatten[{YYrotordisplacement}]}];
XYrotorbladeimpactcoordinates = Transpose[{Flatten[{XXROTORimpactlocation -
       (1.2 * Cfantip * Cos[λtip * P1 / 180] + 1.2 * Cstatorrms * Cos[λstator * P1 / 180] +
         1.2 * Cigvrms * Cos[λigv * P1 / 180]))}], Flatten[{YYROTORimpactlocation}]}];
ListPlot[{XYrotorbladecoordinatesaroundimpact, XYrotorbladeimpactcoordinates},
 GridLines → Automatic,
 PlotLegends → {"XYrotorbladecoordinatesaroundimpact", "XYrotorbladeimpactcoordinates"}];
XYplanerotorbladecoordinatesaroundimpact =
 Transpose[{Flatten[{XXplanerotordisplacement}], Flatten[{YYplanerotordisplacement}]}];
XYparticleplaneimpactcoordinates = Transpose[{Flatten[{XXROTORimpactlocation -
       (1.2 * Cfantip * Cos[λtip * P1 / 180] + 1.2 * Cstatorrms * Cos[λstator * P1 / 180] +
         1.2 * Cigvrms * Cos[λigv * P1 / 180]))}], Flatten[{YYPlaneImpactLocation}]}];
XXYYClosestnodetoparticle = Transpose[{Flatten[{XXClosestnodetoparticle}],
   Flatten[{YYClosestnodetoparticle}]}];
ListPlot[{XYplanerotorbladecoordinatesaroundimpact, XYparticleplaneimpactcoordinates,
  XXYYClosestnodetoparticle}, GridLines → Automatic,
 PlotLegends → {"XYplanerotorbladecoordinatesaroundimpact", "XYparticleplaneimpactcoordinates",
   "XXYYClosestnodetoparticle"}, AxesLabel → SurfaceLocator];
Print[ListPlot[{XYplanerotorbladecoordinatesaroundimpact, XXYYClosestnodetoparticle},
  GridLines → Automatic, PlotLegends → {"XYplanerotorbladecoordinatesaroundimpact",
    "XXYYClosestnodetoparticle"}, AxesLabel → SurfaceLocator]];
Show[Graphics[Circle[{XXClosestnodetoparticle, YYClosestnodetoparticle}, rfod],
  GridLines → Automatic, AxesLabel → SurfaceLocator],
 ListPlot[{XYplanerotorbladecoordinatesaroundimpact, XXYYClosestnodetoparticle},
  GridLines → Automatic, AxesLabel → SurfaceLocator]];
Print["XXClosestnodetoparticle: ", XXClosestnodetoparticle];
```

```
26 |
Print["YYClosestnodetoparticle: ", YYClosestnodetoparticle];
Print["XXYYClosestnodetoparticle: ", XXYYClosestnodetoparticle];
Print["DistanceAtRotorImpact: ", ClosestNode];

(* Graphics routine, very memory process time intensive *)
Print[Show[Legended[Graphics3D[{{PointSize[0.005], Red, Point[FanDisplacementCoordinates]},
     {PointSize[0.005], Red, Point[RotorDisplacementCoordinates]},
     {PointSize[0.007], Green, Point[ParticleDisplacementCoordinates]},
     {Gray, PointSize[0.009], Point[ImpactLocationCoordinates]},
     {Pink, PointSize[0.009], Point[ROTORImpactLocationCoordinates]},
     {Black, PointSize[0.004], Joined → True, Point[StatorLocationCoordinates]},
     {Blue, PointSize[0.009], Point[StatorImpactLocationCoordinates]},
     {Cyan, PointSize[0.004], Joined → True, Point[igvLocationCoordinates]},
     {Blue, PointSize[0.009], Point[igvImpactLocationCoordinates]}}],
   {Red, Style["Fan Displacement", Bold], Red, Style["Rotor Displacement", Bold], Green,
    "Particle Displacement Coordinates", Gray, "Fan Impact Location Coordinates",
    Gray, "Rotor Impact Location Coordinates", Black, "Stator Location",
    Blue, "Stator Impact Location Coordinates", Cyan, "igv Location", Blue,
    "igv Impact Location Coordinates", Transparent, "Engine Duct", Yellow, "Engine Shaft"}],
  ParametricPlot3D[{(u[t] - Cfantip * Cos[λtip * Pi / 180]), -(Dshaft / 4) * (t / (t /.
          Extract[FindMinimum[((u[t] - Cfantip * Cos[λtip * Pi / 180]) /. soln)^2, {t, 0.01}], {2}])) *
      Sin[Theta], -(Dshaft / 4) * (t / (t /. Extract[FindMinimum[((u[t] - Cfantip *
                 Cos[λtip * Pi / 180]) /. soln)^2, {t, 0.01}], {2}])) * Cos[Theta]} /. soln, {t, 0,
   t /. Extract[FindMinimum[((u[t] - Cfantip * Cos[λtip * Pi / 180]) /. soln)^2, {t, 0.001}], {2}]},
  {Theta, 0, Evaluate[2 * Pi]}, PlotRange → {{0, Cfantip * Cos[λtip * Pi / 180]}, {-.5, .5}, {-.5, .5}},
  PlotTheme → {"OpenMarkersThick", "MediumLabels"},
  PlotStyle → {Thickness[0.001], Yellow}],
  ParametricPlot3D[{u[t], -Dshaft / (4 * Cfantip * Cos[λtip * Pi / 180]) *
      (u[t] + Cfantip * Cos[λtip * Pi / 180]) * Sin[Theta], -Dshaft / (4 * Cfantip * Cos[λtip * Pi / 180]) *
      (u[t] + Cfantip * Cos[λtip * Pi / 180]) * Cos[Theta]} /. soln, {t, 0,
   t /. Extract[FindMinimum[((u[t] - Cfantip * Cos[λtip * Pi / 180]) /. soln)^2, {t, 0.01}], {2}]},
  {Theta, 0, Evaluate[2 * Pi]}, PlotRange → {{0, Cfantip * Cos[λtip * Pi / 180]}, {-.5, .5}, {-.5, .5}},
  PlotTheme → {"OpenMarkersThick", "LargeLabels"}, PlotStyle → {Thickness[0.001], Yellow}],
  ParametricPlot3D[{u[t], -Rshaft * Sin[Theta], -Rshaft * Cos[Theta]} /. soln,
   {t, t /. Extract[FindMinimum[((u[t] - Cfantip * Cos[λtip * Pi / 180]) /. soln)^2, {t, 0.001}], {2}],
    t /. Extract[FindMinimum[((u[t] - (1.2 * Cfantip * Cos[λtip * Pi / 180] +
               1.2 * Cstatorrms * Cos[λstator * Pi / 180] + 1.2 * Cigvrms * Cos[λigv * Pi / 180] +
               Cistrotortip * Cos[λistrotortip * Pi / 180])) /. soln)^2, {t, 0.01}], {2}]},
  {Theta, 0, Evaluate[2 * Pi]}, PlotRange → {{Cfantip * Cos[λtip * Pi / 180],
     1.2 * Cfantip * Cos[λtip * Pi / 180] + 1.2 * Cstatorrms * Cos[λstator * Pi / 180] + 1.2 * Cigvrms *
      Cos[λigv * Pi / 180] + Cistrotortip * Cos[λistrotortip * Pi / 180]}, {-.5, .5}, {-.5, .5}},
  PlotTheme → {"OpenMarkersThick", "LargeLabels"}, PlotStyle → {Thickness[0.001], Yellow}],
  ParametricPlot3D[{u[t], -(u[t] * Dfanoutlet * Dinlet * (Cfantip * Cos[λtip * Pi / 180] - u[t])) /
      (2 * Cfantip * Cos[λtip * Pi / 180]) * Sin[Theta],
    -(u[t] * Dfanoutlet * Dinlet * (Cfantip * Cos[λtip * Pi / 180] - u[t])) /
      (2 * Cfantip * Cos[λtip * Pi / 180]) * Cos[Theta]} /. soln, {t, 0,
   t /. Extract[FindMinimum[((u[t] - Cfantip * Cos[λtip * Pi / 180]) /. soln)^2, {t, 0.001}], {2}]},
  {Theta, 0, Evaluate[2 * Pi]}, PlotRange →
   {{-0.001, Cfantip * Cos[λtip * Pi / 180]}, {-.5, .5}, {-.5, .5}},
  PlotTheme → {"OpenMarkersThick", "LargeLabels"}, PlotStyle → {Thickness[0.001], Transparent}],
  ParametricPlot3D[{u[t], -Dstatoroutlet / 2 * Sin[Theta], -Dstatoroutlet / 2 * Cos[Theta]} /. soln,
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)

```
  {t, t /. Extract[FindMinimum[((u[t] - Cfantip * Cos[λtip * P1 / 180]) /. soln)^2, {t, 0.001}], {2}],
   t /. Extract[FindMinimum[((u[t] - (1.2 * Cfantip * Cos[λtip * P1 / 180] +
              Cstatorrms * Cos[λstator * P1 / 180])) /. soln)^2, {t, 0.001}], {2}]},
  {Theta, 0, Evaluate[2 * P1]}, PlotRange → {{Cfantip * Cos[λtip * P1 / 180],
     1.2 * Cfantip * Cos[λtip * P1 / 180] + Cstatorrms * Cos[λstator * P1 / 180]}, {-.5, .5}, {-.5, .5}},
  PlotTheme → {"OpenMarkersThick", "LargeLabels"}, PlotStyle → {Thickness[0.001], Transparent}],
 ParametricPlot3D[{u[t], -Digvoutlet / 2 * Sin[Theta], -Digvoutlet / 2 * Cos[Theta]} /. soln,
  {t, t /. Extract[FindMinimum[((u[t] - (1.2 * Cfantip * Cos[λtip * P1 / 180] +
              Cstatorrms * Cos[λstator * P1 / 180])) /. soln)^2, {t, 0.001}], {2}],
   t /. Extract[FindMinimum[((u[t] - (1.2 * Cfantip * Cos[λtip * P1 / 180] + 1.2 * Cstatorrms *
              Cos[λstator * P1 / 180] + Cigvrms * Cos[λigv * P1 / 180])) /. soln)^2, {t, 0.001}], {2}]},
  {Theta, 0, Evaluate[2 * P1]}, PlotRange → {{1.2 * Cfantip * Cos[λtip * P1 / 180] +
      Cstatorrms * Cos[λstator * P1 / 180], 1.2 * Cfantip * Cos[λtip * P1 / 180] +
     1.2 * Cstatorrms * Cos[λstator * P1 / 180] + 1.2 * Cigvrms * Cos[λigv * P1 / 180]},
    {-.5, .5}, {-.5, .5}}, PlotTheme → {"OpenMarkersThick", "LargeLabels"},
  PlotStyle → {Thickness[0.001], Transparent}], ParametricPlot3D[
  {u[t], -Distrotoroutlet / 2 * Sin[Theta], -Distrotoroutlet / 2 * Cos[Theta]} /. soln,
  {t, t /. Extract[FindMinimum[((u[t] - (1.2 * Cfantip * Cos[λtip * P1 / 180] + 1.2 * Cstatorrms *
              Cos[λstator * P1 / 180] + Cigvrms * Cos[λigv * P1 / 180])) /. soln)^2, {t, 0.001}], {2}],
   t /. Extract[FindMinimum[((u[t] - (1.2 * Cfantip * Cos[λtip * P1 / 180] + 1.2 * Cstatorrms *
              Cos[λstator * P1 / 180] + 1.2 * Cigvrms * Cos[λigv * P1 / 180] +
             Cistrotortip * Cos[λistrotortip * P1 / 180])) /. soln)^2, {t, 0.001}], {2}]},
  {Theta, 0, Evaluate[2 * P1]}, PlotRange → {{1.2 * Cfantip * Cos[λtip * P1 / 180] +
      1.2 * Cstatorrms * Cos[λstator * P1 / 180] + 1.2 * Cigvrms * Cos[λigv * P1 / 180],
     1.2 * Cfantip * Cos[λtip * P1 / 180] + 1.2 * Cstatorrms * Cos[λstator * P1 / 180] +
      1.2 * Cigvrms * Cos[λigv * P1 / 180] + Cistrotortip * Cos[λistrotortip * P1 / 180]},
    {-.5, .5}, {-.5, .5}}, PlotTheme → {"OpenMarkersThick", "LargeLabels"},
  PlotStyle → {Thickness[0.001], Transparent}, Axes → True]];

Xcg = 0;
Ycg = 0;
Do[Xcg = Xcg + XXplanerotordisplacement[[1]];
 Ycg = Ycg + YYplanerotordisplacement[[1]], {1, 1, 51, 1}];
XCG = Xcg / 51;
YCG = Ycg / 51;
Print["XCG: ", XCG, " YCG: ", YCG];
Print["Axialσ: ", Axialσ];

If[SurfaceLocator == "Suction Side" && Rimpact > 0, Print["VSuctionSide: ", VSuctionSide];
 XYLocationofImpactonSuctionSiderotor = AppendTo[
   XYLocationofImpactonSuctionSiderotor, {XXClosestnodetoparticle, YYClosestnodetoparticle}];
 ChordSuctionSiderotor = Chord;
 Print["ChordSuctionSiderotor: ", ChordSuctionSiderotor];
 XXPercentChordLocationSuctionSiderotor =
  AppendTo[XXPercentChordLocationSuctionSiderotor, XpercentChord];
 VSuctionSideAppended = AppendTo[VSuctionSideAppended, VSuctionSide];
 VSuctionSideAppendedTotal = AppendTo[VSuctionSideAppendedTotal, VSuctionSide];
 RimpactAppendedSuctionSiderotor = AppendTo[RimpactAppendedSuctionSiderotor, Rimpact];
 RimpactAppendedSuctionSiderotorTotal = AppendTo[RimpactAppendedSuctionSiderotorTotal, Rimpact];
 XXPercentChordLocationSuctionSiderotorTotal =
  AppendTo[XXPercentChordLocationSuctionSiderotorTotal, XpercentChord];
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)

```
28 |
VSuctionSidevsChordLocationAppended = AppendTo[VSuctionSidevsChordLocationAppended,
  {XpercentChord, VSuctionSide}];
VSuctionSidevsChordLocationAppendedTotal = If[VSuctionSide > 0,
  AppendTo[VSuctionSidevsChordLocationAppendedTotal, {XpercentChord, VSuctionSide}]];
RimpactVsXpctAppendedSuctionSiderotor = AppendTo[
  RimpactVsXpctAppendedSuctionSiderotor, {Rimpact, XpercentChord}],
If[SurfaceLocator == "Pressure Side" && Rimpact > 0, Print["VPressureSide: ", VPressureSide];
 XYLocationofImpactonPressureSiderotor = AppendTo[
   XYLocationofImpactonPressureSiderotor, {XXClosestnodetoparticle, YYClosestnodetoparticle}];
 ChordPressureSiderotor = Chord;
 Print["ChordPressureSiderotor: ", ChordPressureSiderotor];
 XXPercentChordLocationPressureSiderotor =
  AppendTo[XXPercentChordLocationPressureSiderotor, XpercentChord];
 VPressureSideAppended = AppendTo[VPressureSideAppended, VPressureSide];
 VPressureSideAppendedTotal = AppendTo[VPressureSideAppendedTotal, VPressureSide];
 RimpactAppendedPressureSiderotor = AppendTo[RimpactAppendedPressureSiderotor, Rimpact];
 RimpactAppendedPressureSiderotorTotal =
  AppendTo[RimpactAppendedPressureSiderotorTotal, Rimpact];
 XXPercentChordLocationPressureSiderotorTotal =
  AppendTo[XXPercentChordLocationPressureSiderotorTotal, XpercentChord];
 VPressureSidevsChordLocationAppended = AppendTo[VPressureSidevsChordLocationAppended,
   {XpercentChord, VPressureSide}];
 VPressureSidevsChordLocationAppendedTotal = If[VPressureSide > 0,
   AppendTo[VPressureSidevsChordLocationAppendedTotal, {XpercentChord, VPressureSide}]];
 RimpactVsXpctAppendedPressureSiderotor = AppendTo[
   RimpactVsXpctAppendedPressureSiderotor, {Rimpact, XpercentChord}]]];
If[SurfaceLocator == "Suction Side" && Rimpact > 0, nimpactsuctionsiderotor =
  nimpactsuctionsiderotor + 1, If[SurfaceLocator == "Pressure Side" && Rimpact > 0,
  nimpactpressuresiderotor = nimpactpressuresiderotor + 1]];


(*Module for Impact Depth and Diameter*)

Vimpact = If[SurfaceLocator == "Suction Side",
  VSuctionSide, If[SurfaceLocator == "Pressure Side", VPressureSide, 0]];
ChordatImpact = If[SurfaceLocator == "Suction Side", ChordSuctionSiderotor,
  If[SurfaceLocator == "Pressure Side", ChordPressureSiderotor, 0]];

μGlassPenetrationDepthfor2mmImpactor = 0.000598;
σGlassPenetrationDepthfor2mmImpactor = 0.000183;
μGlassImpactVelocityfor2mmImpactor = 304.8;
σGlassImpactVelocityfor2mmImpactor = 6.7;

μGlassPenetrationDepthfor5mmImpactor = 0.000745;
σGlassPenetrationDepthfor5mmImpactor = 0.000107;
μGlassImpactVelocityfor5mmImpactor = 295.7;
σGlassImpactVelocityfor5mmImpactor = 6.7;

μSteelPenetrationDepthfor133mmImpactor = 0.0003515;
σSteelPenetrationDepthfor133mmImpactor = 0.000229848;
μSteelImpactVelocityfor133mmImpactor = 267.9482599;
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)


```
σSteelImpactVelocityfor133mmImpactor = 148.0193977;

μSteelPenetrationDepthfor2mmImpactor = 0.000717597;
σSteelPenetrationDepthfor2mmImpactor = 0.000108401;
μSteelImpactVelocityfor2mmImpactor = 310.7320881;
σSteelImpactVelocityfor2mmImpactor = 7.703985571;

μSteelPenetrationDepthfor32mmImpactor = 0.0004982;
σSteelPenetrationDepthfor32mmImpactor = 0.000151357;
μSteelImpactVelocityfor32mmImpactor = 226.6025404;
σSteelImpactVelocityfor32mmImpactor = 48.89090684;

μSteelPenetrationDepthfor4mmImpactor = 0.001868;
σSteelPenetrationDepthfor4mmImpactor = 0.000315;
μSteelImpactVelocityfor4mmImpactor = 306.5;
σSteelImpactVelocityfor4mmImpactor = 9.2;

(*Choose Material: 1 for Glass or 2 for Steel*)
ImpactorMaterial = 2;
If[Vimpact > 0,
 If[ImpactorMaterial == 1,
  (*Choose Particle Diameter (if Steel choose 1 for 1.33 mm, 2 for 2 mm,
    3 for 3.2 mm or 4 for 4 mm, if Glass choose 1 for 2 mm or 2 for 5 mm)*)
  ParticleSize = 1;
  If[ParticleSize == 1,
   ProbGivenVimpactNormalGlass2mm = CDF[NormalDistribution[
      μGlassImpactVelocityfor2mmImpactor, σGlassImpactVelocityfor2mmImpactor], Vimpact];
   distGlass2mm = NormalDistribution[μGlassPenetrationDepthfor2mmImpactor,
     σGlassPenetrationDepthfor2mmImpactor];
   sol2glass2mm = FindRoot[ProbGivenVimpactNormalGlass2mm ==
      Probability[x ≤ aa, Distributed[x, distGlass2mm]],
     {aa, μGlassPenetrationDepthfor2mmImpactor + σGlassPenetrationDepthfor2mmImpactor}];
   DepthofPenetration = If[Evaluate[aa /. sol2glass2mm] ≤ 0.000001, 0, aa /. sol2glass2mm];
   Print["DepthofPenetration: ", DepthofPenetration],
   If[ParticleSize == 2,
    ProbGivenVimpactNormalGlass5mm = CDF[NormalDistribution[
       μGlassImpactVelocityfor5mmImpactor, σGlassImpactVelocityfor5mmImpactor], Vimpact];
    distGlass5mm = NormalDistribution[μGlassPenetrationDepthfor5mmImpactor,
      σGlassPenetrationDepthfor5mmImpactor];
    sol2glass5mm = FindRoot[ProbGivenVimpactNormalGlass5mm ==
       Probability[x ≤ aa, Distributed[x, distGlass5mm]],
      {aa, μGlassPenetrationDepthfor5mmImpactor + σGlassPenetrationDepthfor5mmImpactor}];
    DepthofPenetration = If[Evaluate[aa /. sol2glass5mm] ≤ 0.000001, 0, aa /. sol2glass5mm];
    Print["DepthofPenetration: ", DepthofPenetration],
    Print["Choose from Glass or Steel and indentor diameter and re-run"];
    Abort[]]],
  If[ImpactorMaterial == 2,
   ParticleSize = 3;
   If[ParticleSize == 1,
    FodDiam = 0.00133;
    ProbGivenVimpactNormalSteel133mm = CDF[NormalDistribution[
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)

```
30 |
      μSteelImpactVelocityfor133mmImpactor, σSteelImpactVelocityfor133mmImpactor], Vimpact];
distSteel133mm = NormalDistribution[μSteelPenetrationDepthfor133mmImpactor,
  σSteelPenetrationDepthfor133mmImpactor];
sol2steel133mm = FindRoot[ProbGivenVimpactNormalSteel133mm ==
    Probability[x ≤ aa, Distributed[x, distSteel133mm]],
  {aa, μSteelPenetrationDepthfor133mmImpactor + σSteelPenetrationDepthfor133mmImpactor}];
DepthofPenetration = If[Evaluate[aa /. sol2steel133mm] ≤ 0.000001, 0, aa /. sol2steel133mm];
PenetrationDiam =
 2 * If[DepthofPenetration ≤ FodDiam/2, ((FodDiam/2)^2 - (FodDiam/2 - DepthofPenetration)^2)^
    0.5, ((FodDiam/2)^2 - (FodDiam/2 - FodDiam/2)^2)^0.5];
Print["DepthofPenetration: ", DepthofPenetration];
Print["PenetrationDiam: ", PenetrationDiam],
If[ParticleSize == 2,
 FodDiam = 0.002;
 ProbGivenVimpactNormalSteel2mm = CDF[NormalDistribution[
    μSteelImpactVelocityfor2mmImpactor, σSteelImpactVelocityfor2mmImpactor], Vimpact];
 distSteel2mm = NormalDistribution[μSteelPenetrationDepthfor2mmImpactor,
   σSteelPenetrationDepthfor2mmImpactor];
 sol2steel2mm = FindRoot[ProbGivenVimpactNormalSteel2mm ==
    Probability[x ≤ aa, Distributed[x, distSteel2mm]],
   {aa, μSteelPenetrationDepthfor2mmImpactor + σSteelPenetrationDepthfor2mmImpactor}];
 DepthofPenetration = If[Evaluate[aa /. sol2steel2mm] ≤ 0.000001, 0, aa /. sol2steel2mm];
 PenetrationDiam = 2 * If[DepthofPenetration ≤ FodDiam/2, ((FodDiam/2)^2 - (FodDiam/2 -
         DepthofPenetration)^2)^0.5, ((FodDiam/2)^2 - (FodDiam/2 - FodDiam/2)^2)^0.5];
 Print["DepthofPenetration: ", DepthofPenetration];
 Print["PenetrationDiam: ", PenetrationDiam],
 If[ParticleSize == 3,
  FodDiam = 0.0032;
  ProbGivenVimpactNormalSteel32mm = CDF[NormalDistribution[
     μSteelImpactVelocityfor32mmImpactor, σSteelImpactVelocityfor32mmImpactor], Vimpact];
  distSteel32mm = NormalDistribution[μSteelPenetrationDepthfor32mmImpactor,
    σSteelPenetrationDepthfor32mmImpactor];
  sol2steel32mm = FindRoot[ProbGivenVimpactNormalSteel32mm ==
     Probability[x ≤ aa, Distributed[x, distSteel32mm]],
    {aa, μSteelPenetrationDepthfor32mmImpactor + σSteelPenetrationDepthfor32mmImpactor}];
  DepthofPenetration = If[Evaluate[aa /. sol2steel32mm] ≤ 0.000001, 0, aa /. sol2steel32mm];
  PenetrationDiam = 2 * If[DepthofPenetration ≤ FodDiam/2, ((FodDiam/2)^2 - (FodDiam/2 -
          DepthofPenetration)^2)^0.5, ((FodDiam/2)^2 - (FodDiam/2 - FodDiam/2)^2)^0.5];
  Print["DepthofPenetration: ", DepthofPenetration];
  Print["PenetrationDiam: ", PenetrationDiam],
  If[ParticleSize == 4,
   FodDiam = 0.004;
   ProbGivenVimpactNormalSteel4mm = CDF[NormalDistribution[
      μSteelImpactVelocityfor4mmImpactor, σSteelImpactVelocityfor4mmImpactor], Vimpact];
   distSteel4mm = NormalDistribution[μSteelPenetrationDepthfor4mmImpactor,
     σSteelPenetrationDepthfor4mmImpactor];
   sol2steel4mm = FindRoot[ProbGivenVimpactNormalSteel4mm ==
      Probability[x ≤ aa, Distributed[x, distSteel4mm]],
     {aa, μSteelPenetrationDepthfor4mmImpactor + σSteelPenetrationDepthfor4mmImpactor}];
   DepthofPenetration = If[Evaluate[aa /. sol2steel4mm] ≤ 0.000001, 0, aa /. sol2steel4mm];
   PenetrationDiam = 2 * If[DepthofPenetration ≤ FodDiam/2, ((FodDiam/2)^2 - (FodDiam/2 -
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)

```
                DepthofPenetration)^2)^0.5, ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5]; 31
      Print["DepthofPenetration: ", DepthofPenetration];
      Print["PenetrationDiam: ", PenetrationDiam],
      Print["Choose from Glass or Steel and indentor diameter and re-run"];
      Abort[]]]]],
   Print["Choose from Glass or Steel and re-run"];
   Abort[]]], DepthofPenetration = 0;
 PenetrationDiam = 0];

(* Routine to calculate Betas at impact for impact stress intensity,
not used as crack growth simulation performs this
  ImpactEccentricty=ChordatImpact*Abs[XpercentChord-0.5];
AverageAirfoilThickness=0.0551*ChordatImpact;

If[PenetrationDiam>0,
 If[Abs[ImpactEccentricty-ChordatImpact/2]/ChordatImpact≥0.03,
  BetaCFC=If[PenetrationDiam>0, ((DepthofPenetration/(PenetrationDiam/2))^0.5)/
     NIntegrate[(1-(1-(DepthofPenetration/(PenetrationDiam/2))^2)*(Sin[φ])^2), {φ, 0, P1/2}], 0];
  BetaBFS=1.12;
  G1=If[PenetrationDiam>0,ArcCos[PenetrationDiam/(ChordatImpact-2*ImpactEccentricty)], 0];
  G2=If[PenetrationDiam>0,ArcCos[PenetrationDiam/(ChordatImpact+2*ImpactEccentricty)], 0];
  G3=If[PenetrationDiam>0,
    Log[(ChordatImpact-ImpactEccentricty+((ChordatImpact/2-ImpactEccentricty)^2-
          (PenetrationDiam/2)^2)^0.5)/(PenetrationDiam/2)], 0];
  G4=If[PenetrationDiam>0, Log[(ChordatImpact+ImpactEccentricty+((ChordatImpact/2+
          ImpactEccentricty)^2-(PenetrationDiam/2)^2)^0.5)/(PenetrationDiam/2)], 0];
  BetaW=If[PenetrationDiam>0, P1*G4/(G1*G4+G2*G3), 0];
  BetaFFS=BetaFFS=If[PenetrationDiam>0,
     1.08882991825178+0.191072321954091*(DepthofPenetration/AverageAirfoilThickness)+
      -0.2275*(DepthofPenetration/PenetrationDiam)+
      ((DepthofPenetration/AverageAirfoilThickness)-0.2375)*
       (((DepthofPenetration/AverageAirfoilThickness)-0.2375)*-0.336251716696097)+
      ((DepthofPenetration/AverageAirfoilThickness)-0.2375)*
       (((DepthofPenetration/PenetrationDiam)-0.256666666666667)*-0.642299410757611)+
      ((DepthofPenetration/PenetrationDiam)-0.256666666666667)*
       (((DepthofPenetration/PenetrationDiam)-0.256666666666667)*0.443509615384615), 0];
  BetaCompounded=BetaBFS*BetaFFS*BetaW*BetaCFC, BetaCompounded=
   1.12-0.231*(DepthofPenetration/ChordatImpact)+
    10.550*(DepthofPenetration/ChordatImpact)^2-21.710*(DepthofPenetration/ChordatImpact)^3+
    30.382*(DepthofPenetration/ChordatImpact)^4], BetaCompounded=0];
*)

NimpactAppended = AppendTo[NimpactAppended, Nimpact];
LocalThicknessatImpactAppended =
 AppendTo[LocalThicknessatImpactAppended, LocalThicknessatImpact];
RIMPACTRotorAppend = AppendTo[RIMPACTRotorAppend, Rimpact];
XpercentChordAppend = AppendTo[XpercentChordAppend, XpercentChord];
ImpactSideRotorAppend = AppendTo[ImpactSideRotorAppend, SurfaceLocator];
ImpactVelocityatRotor = AppendTo[ImpactVelocityatRotor, Vimpact];
StressatImpactSiteAppend = AppendTo[StressatImpactSiteAppend, Axialσ];
DepthofPenetrationAppend = AppendTo[DepthofPenetrationAppend, DepthofPenetration];
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)

```
32 | DiametorofPenetrationAppend = AppendTo[DiametorofPenetrationAppend, PenetrationDiam];
   ChordatImpactAppend = AppendTo[ChordatImpactAppend, ChordatImpact];
   BetaCFCAppend = AppendTo[BetaCFCAppend, BetaCFC];
   BetaWAppend = AppendTo[BetaWAppend, BetaW];
   BetaFFSAppend = AppendTo[BetaFFSAppend, BetaFFS];
   BetaCompoundedAppend = AppendTo[BetaCompoundedAppend, BetaCompounded];



   Rimpact = 0;
   XpercentChord = 0;
   SurfaceLocator = 0;
   VSuctionSide = 0;
   VPressureSide = 0;
   Axialσ = 0;
   DepthofPenetration = 0;
   PenetrationDiam = 0;
   ChordatImpact = 0;
   BetaCFC = 0;
   ImpactEccentricty = 0;
   AverageAirfoilThickness = 0;
   G1 = 0;
   G2 = 0;
   G3 = 0;
   G4 = 0;
   BetaW = 0;
   BetaFFS = 0;
   BetaCompounded = 0;
   LocalThicknessatImpact = 0;
   Nimpact = 0;


   Print[Style["Parameters
       RIMPACTRotorAppend, XpercentChordAppend, ImpactSideRotorAppend, ImpactVelocityatRotor,
       StressatImpactSiteAppend, DepthofPenetrationAppend, DiametorofPenetrationAppend,
       ChordatImpactAppend, BetaCompoundedAppend, NimpactAppended, LocalThicknessatImpactAppended:
        ", FontSize → 24, Black, Bold], RIMPACTRotorAppend,
    XpercentChordAppend, ImpactSideRotorAppend, ImpactVelocityatRotor, StressatImpactSiteAppend,
    DepthofPenetrationAppend, DiametorofPenetrationAppend, ChordatImpactAppend,
    BetaCompoundedAppend, NimpactAppended, LocalThicknessatImpactAppended];
   ParametersAppended = Transpose[{RIMPACTRotorAppend, XpercentChordAppend,
      ImpactSideRotorAppend, ImpactVelocityatRotor, StressatImpactSiteAppend,
      DepthofPenetrationAppend, DiametorofPenetrationAppend, ChordatImpactAppend,
      BetaCompoundedAppend, NimpactAppended, LocalThicknessatImpactAppended}];


   ClearAll[XXYYClosestnodetoparticle, SurfaceLocator],
   {p, 1, Nruns}];
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)


```
MeanRimpactAppendedSuctionSiderotor =
 AppendTo[MeanRimpactAppendedSuctionSiderotor, Mean[RimpactAppendedSuctionSiderotor]];
STDRimpactAppendedSuctionSiderotor = AppendTo[STDRimpactAppendedSuctionSiderotor,
  StandardDeviation[RimpactAppendedSuctionSiderotor]];
MeanRimpactAppendedPressureSiderotor = AppendTo[MeanRimpactAppendedPressureSiderotor,
  Mean[RimpactAppendedPressureSiderotor]];
STDRimpactAppendedPressureSiderotor = AppendTo[STDRimpactAppendedPressureSiderotor,
  StandardDeviation[RimpactAppendedPressureSiderotor]];
MeanVSuctionSide = Mean[VSuctionSideAppended];
MeanVPressureSide = Mean[VPressureSideAppended];
MeanVSuctionSideAppended = AppendTo[MeanVSuctionSideAppended, Mean[VSuctionSideAppended]];
MeanVPressureSideAppended = AppendTo[MeanVPressureSideAppended, Mean[VPressureSideAppended]];
STDVSuctionSide = StandardDeviation[VSuctionSideAppended];
STDVPressureSide = StandardDeviation[VPressureSideAppended];
STDVSuctionSideAppended =
 AppendTo[STDVSuctionSideAppended, StandardDeviation[VSuctionSideAppended]];
STDVPressureSideAppended = AppendTo[STDVPressureSideAppended,
  StandardDeviation[VPressureSideAppended]];
MeanXXPercentChordLocationSuctionSiderotor = AppendTo[MeanXXPercentChordLocationSuctionSiderotor,
  Mean[XXPercentChordLocationSuctionSiderotor]];
MeanXXPercentChordLocationPressureSiderotor = AppendTo[
  MeanXXPercentChordLocationPressureSiderotor, Mean[XXPercentChordLocationPressureSiderotor]];
STDXXPercentChordLocationSuctionSiderotor = AppendTo[STDXXPercentChordLocationSuctionSiderotor,
  StandardDeviation[XXPercentChordLocationSuctionSiderotor]];
STDXXPercentChordLocationPressureSiderotor = AppendTo[STDXXPercentChordLocationPressureSiderotor,
  StandardDeviation[XXPercentChordLocationPressureSiderotor]];


Print["nimpactsuctionsiderotor: ", nimpactsuctionsiderotor];
Print["XYLocationofImpactonSuctionSiderotor: ", XYLocationofImpactonSuctionSiderotor];
Print["nimpactpressuresiderotor: ", nimpactpressuresiderotor];
Print["XYLocationofImpactonPressureSiderotor: ", XYLocationofImpactonPressureSiderotor];


(*XXPercentChordLocationSuctionSiderotor
 Transpose[{Flatten[{XXPercentChordLocationSuctionSiderotor}], Flatten[{VSuctionSideAppended}]}]
 Transpose[
  {Flatten[{XXPercentChordLocationPressureSiderotor}], Flatten[{VPressureSideAppended}]}]*)


If[XYLocationofImpactonSuctionSiderotor ≠ {} && XYLocationofImpactonPressureSiderotor ≠ {},
 Show[ListPlot[{XYLocationofImpactonSuctionSiderotor, XYLocationofImpactonPressureSiderotor},
   GridLines → Automatic, PlotLegends → {"XYLocationofImpactonSuctionSiderotor",
     "XYLocationofImpactonPressureSiderotor"}], Axes → True],
 If[XYLocationofImpactonSuctionSiderotor ≠ {} && XYLocationofImpactonPressureSiderotor == {},
  Print[ListPlot[XYLocationofImpactonSuctionSiderotor, GridLines → Automatic,
    PlotLegends → {"XYLocationofImpactonSuctionSiderotor"}]],
  If[XYLocationofImpactonSuctionSiderotor == {} && XYLocationofImpactonPressureSiderotor ≠ {},
   Print[ListPlot[XYLocationofImpactonPressureSiderotor, GridLines → Automatic,
     PlotLegends → {"XYLocationofImpactonPressureSiderotor"}]]]];
```

34 |

```
Print[ListPlot[{VSuctionSidevsChordLocationAppended,
   VPressureSidevsChordLocationAppended}, GridLines → Automatic,
  PlotLegends → {"VSuctionSidevsChordLocationAppended", "VPressureSidevsChordLocationAppended"},
  AxesLabel → {"% Chord", "Impact Velocity"}]];

Print["ChordSuctionSiderotor: ", ChordSuctionSiderotor];
Print["XXPercentChordLocationSuctionSiderotor: ", XXPercentChordLocationSuctionSiderotor];
Print["ChordPressureSiderotor: ", ChordPressureSiderotor];
Print["XXPercentChordLocationPressureSiderotor: ", XXPercentChordLocationPressureSiderotor];


Print[Histogram[{XXPercentChordLocationSuctionSiderotor, XXPercentChordLocationPressureSiderotor},
  Max[Length[XXPercentChordLocationSuctionSiderotor],
    Length[XXPercentChordLocationPressureSiderotor]] + 1, ChartLegends →
   {"XXPercentChordLocationSuctionSiderotor", "XXPercentChordLocationPressureSiderotor"},
  ChartStyle → "Pastel", AxesLabel → {"% Chord", "Frequency"}]];

Print[Histogram[{VSuctionSideAppended, VPressureSideAppended},
  Max[Length[VSuctionSideAppended], Length[VPressureSideAppended]] + 1,
  ChartLegends → {"VSuctionSideAppended", "VPressureSideAppended"},
  ChartStyle → "Pastel", AxesLabel → {"Impact Velocity", "Frequency"}]];


VSuctionSidevsChordLocationAppended = {};
VPressureSidevsChordLocationAppended = {};

ClearAll[XYrotorbladecoordinatesaroundimpact, ImpactLocationCoordinates,
 FanDisplacementCoordinates, RotorImpactLocationCoordinates, RotorDisplacementCoordinates,
 NextFanDisplacementCoordinates, PreviousFanDisplacementCoordinates,
 XYrotorbladeimpactcoordinates, XYplanerotorbladecoordinatesaroundimpact,
 XYparticleplaneimpactcoordinates, XYplaneimpactOML, VPressureSide, VSuctionSide, Axialσ],

{k, 1, InputFileLength}]

SteelParticleFullFactorialOutput = ParametersAppended;
Print["SteelParticleFullFactorialOutput: ", SteelParticleFullFactorialOutput];
Print[MatrixForm[SteelParticleFullFactorialOutput,
   TableHeadings → {None, {"RIMPACT", "XpercentChord", "ImpactSideRotor", "ImpactVelocityatRotor",
      "StressatImpactSite", "DepthofPenetration", "DiameterofPenetration",
      "ChordatImpact", "BetaCompounded", "Nimpact", "LocalThicknessatImpact"}}]];

Export[
 "C:\\Users\\mg151\\MathematicaRuns\\InEngineParticleMotionNoBoundaryLayerHeadWindandURollingGrou
    ndEqual32mmSteelInput4FactorsOutput1081Runs.csv", SteelParticleFullFactorialOutput, "CSV"]

Print["RimpactVsXpctAppendedSuctionSiderotor: ", RimpactVsXpctAppendedSuctionSiderotor]
Print["RimpactVsXpctAppendedPressureSiderotor: ", RimpactVsXpctAppendedPressureSiderotor]

Print["VSuctionSidevsChordLocationAppendedTotal: ", VSuctionSidevsChordLocationAppendedTotal]
Print["VPressureSidevsChordLocationAppendedTotal: ", VPressureSidevsChordLocationAppendedTotal]
```

Printed by Wolfram Mathematica Student Edition

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (continued)


```
Histogram3D[RimpactVsXpctAppendedSuctionSiderotor,
 {Length[RimpactVsXpctAppendedSuctionSiderotor] + 1, {0.01}},
 AxesLabel → {"R", "% Chord", "Frequency"}, ChartStyle → "Pastel",
 ChartLabels → Placed[{"Histogram of Radius at Impact and % Chord Suction Side"}, Bottom]]
Histogram3D[RimpactVsXpctAppendedPressureSiderotor,
 {Length[RimpactVsXpctAppendedPressureSiderotor] + 1, {0.01}},
 AxesLabel → {"R", "% Chord", "Frequency"}, ChartStyle → "Pastel",
 ChartLabels → Placed[{"Histogram of Radius at Impact and % Chord Pressure Side"}, Bottom]]
Histogram3D[VSuctionSidevsChordLocationAppendedTotal,
 {Length[VSuctionSidevsChordLocationAppendedTotal] + 1, {0.01}},
 AxesLabel → {"% Chord", "Impact Velocity on Suction Side", "Frequency"}, ChartStyle → "Pastel",
 ChartLabels → Placed[{"Histogram of Impact Velocity and % Chord Suction Side"}, Bottom]]
Histogram3D[VPressureSidevsChordLocationAppendedTotal,
 {Length[VPressureSidevsChordLocationAppendedTotal] + 1, {0.01}},
 AxesLabel → {"% Chord", "Impact Velocity on Pressure Side", "Frequency"}, ChartStyle → "Pastel",
 ChartLabels → Placed[{"Histogram of Impact Velocity and % Chord Pressure Side"}, Bottom]]

Print["RimpactAppendedSuctionSiderotorTotal: ", RimpactAppendedSuctionSiderotorTotal]
Print["RimpactAppendedPressureSiderotorTotal: ", RimpactAppendedPressureSiderotorTotal]
Print["XXPercentChordLocationSuctionSiderotorTotal: ",
 XXPercentChordLocationSuctionSiderotorTotal]
Print["XXPercentChordLocationPressureSiderotorTotal: ",
 XXPercentChordLocationPressureSiderotorTotal]
MeanRimpactAppendedSuctionSiderotorTotal = Mean[RimpactAppendedSuctionSiderotorTotal];
MeanRimpactAppendedPressureSiderotorTotal = Mean[RimpactAppendedPressureSiderotorTotal];
STDRimpactAppendedSuctionSiderotorTotal = StandardDeviation[RimpactAppendedSuctionSiderotorTotal];
STDRimpactAppendedPressureSiderotorTotal =
  StandardDeviation[RimpactAppendedPressureSiderotorTotal];
MeanXXPercentChordLocationSuctionSiderotorTotal =
  Mean[XXPercentChordLocationSuctionSiderotorTotal];
MeanXXPercentChordLocationPressureSiderotorTotal =
  Mean[XXPercentChordLocationPressureSiderotorTotal];
STDXXPercentChordLocationSuctionSiderotorTotal =
  StandardDeviation[XXPercentChordLocationSuctionSiderotorTotal];
STDXXPercentChordLocationPressureSiderotorTotal =
  StandardDeviation[XXPercentChordLocationPressureSiderotorTotal];
CovarianceRimpactXXPercentChordLocationSuctionSiderotorTotal =
 Covariance[RimpactAppendedSuctionSiderotorTotal, XXPercentChordLocationSuctionSiderotorTotal]
CovarianceRimpactXXPercentChordLocationPressureSiderotorTotal =
 Covariance[RimpactAppendedPressureSiderotorTotal, XXPercentChordLocationPressureSiderotorTotal]

Plot3D[PDF[BinormalDistribution[
   {MeanRimpactAppendedSuctionSiderotorTotal, MeanXXPercentChordLocationSuctionSiderotorTotal},
   {STDRimpactAppendedSuctionSiderotorTotal, STDXXPercentChordLocationSuctionSiderotorTotal},
   CovarianceRimpactXXPercentChordLocationSuctionSiderotorTotal], {Radius, PctChord}}],
 {Radius, 0.21, 0.38}, {PctChord, 0, 1}, PlotRange → All, MeshFunctions → {#3 &},
 MeshShading → {None, Red, None, Yellow}, PlotPoints → 25, AxesLabel → Automatic,
 PlotLabel → "Binormal Distribution of Radius vs. % Chord Suction Side"]

Plot3D[PDF[BinormalDistribution[
```

In-Engine Particle Kinetics Code – 3.2 mm Steel Debris (complete)

```
36 |
    {MeanRimpactAppendedPressureSiderotorTotal, MeanXXPercentChordLocationPressureSiderotorTotal},
    {STDRimpactAppendedPressureSiderotorTotal, STDXXPercentChordLocationPressureSiderotorTotal},
    CovarianceRimpactXXPercentChordLocationPressureSiderotorTotal}, {Radius, PctChord}],
  {Radius, 0.21, 0.38}, {PctChord, 0, 1}, PlotRange → All, MeshFunctions → (#3 &),
  MeshShading → {None, Red, None, Yellow}, PlotPoints → 25, AxesLabel → Automatic,
  PlotLabel → "Binormal Distribution of Radius vs. % Chord Pressure Side"]

Print["NrunsCount: ", NrunsCount]

Print["NfanstrikesTotal: ", NfanstrikesTotal]
Print["NstatorstrikesTotal: ", NstatorstrikesTotal]
Print["NigvstrikesTotal: ", NigvstrikesTotal]
Print["NrotorstrikesTotal: ", NrotorstrikesTotal]
Print["NshaftstrikesTotal: ", NshaftstrikesTotal]
Print["NbypassexitTotal: ", NbypassexitTotal]
Print["NtaperedfanhousingstrikesTotal: ", NtaperedfanhousingstrikesTotal]
```

APPENDIX F: CRACK GROWTH AND LCC ESTIMATION

Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD

```
ClearSystemCache
Off[Power::infy]
"Statistics`NonlinearFit`"
AppendTo[$Path,
  ToFileName[{$HomeDirectory, "GeorgiaTech\\Research\\MyResearch\\MathematicaRuns"}]];
CrackGrowthModel = Import[
  "CrackGrowthModelNoBoundaryLayerHeadWindandUGEqual133mmSteelInput_FullFactorialAutomatedInspect\
    ions72Runs.csv", "CSV"]
InputFileLength = Length[CrackGrowthModel];
TotalLifeTimeCostsAppendedDOE = {};
NrepairsAppendedDOE = {};
NreplacementsAppendedDOE = {};
aacAppended = {};
NEnginereplacementsAppendedDOE = {};
NimpactsAppendedDOE = {};
TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisitionAppendedDOE = {};
CAMATreplacementsAppendedDOE = {};
LifeTimerepairscostsAppendedDOE = {};
LifeTimereplacementscostsAppendedDOE = {};
TotalLifeTimeCostsAppendedDOEAutomatedMean = {};
TotalLifeTimeCostsAppendedDOEAutomatedStd = {};
TotalLifeTimeCostsAppendedDOEAutomatedMeanvsDOESetting = {};
TotalLifeTimeCostsAppendedDOEAutomatedStdvsDOESetting = {};
NreplacementsvsDOESettingAppendedAutomatedSum = {};
NrepairsvsDOESettingAppendedAutomatedSum = {};

Do[

 NbladereplacementsAppended = {};
 Ncyclesatbladereplacement = {};
 aaOregressedAppended = {};
 NcyclesatbladereplacementTotal = {};
 Nimpacts = 0;
 Nrepairs = 0;
 Nreplacements = 0;
 NrepairsAppended = {};
 NreplacementsAppended = {};
 SCHCALrepairsAppended = {};
 LreplacementsAppended = {};
 SCHEDCAMATrepairsAppended = {};
 CAMATreplacementsAppended = {};
 TotalLifeTimeCostsAppended = {}; NEnginereplacements = 0;
 NEnginereplacementsAppended = {};
 LEnginereplacementsAppended = {};
 MATEnginereplacementsAppended = {};
 NimpactsAppended = {};
 LifeTimeFuelCostDueToEfficiencyDegradation = 0;
 TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisitionAppended = {};
 LifeTimerepairscostsAppended = {};
 LifeTimereplacementscostsAppended = {};
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

2 |

```
LifeTimeRepairs = 0;
LifeTimeReplacements = 0;
SCHCALRepairs = 0;
LReplacements = 0;
LEngineReplacements = 0;
LifeTimeEngineReplacements = 0;
LifeTimeRepairsCosts = 0;
LifeTimeReplacementsCosts = 0;

(* Runway Geometry, FOD Quantity on Runway,
FOD Density on Runway, FOD Geometry, FOD Sweeping performance factors *)
Lrunway = 3048; (* Runway Length *)
Wrunway = 76.2; (* Runway Width *)
FODCountonRunway = 25 000;
(* Ammount of continuously/evenly distributed FOD on runway *)
ρFodonRunway = 1 / 9.2903;
(* Density of continuously/evenly distributed FOD on runway *)
DParticle = 0.00133;
Dstagpoint = 0.0030 * 2 * (30)^0.5;
(* Scaled diameter of vortex stagnation point on runway *)
nrotorblades = 38;
(* Number of rotor blades on the 1st stage compressor rotor *)
Ninspections = 2; (* Number of runway visual inspections *)
ηvisual = 0.80; (* Current efficacy of visual inspections at detecting FOD on the runway *)
ηsweep = 0.90; (* Current efficacy of FOD removal from sweeper on runway *)
FODBossSweptAreaperhour = 300 000;
(* FOD-Boss in the Triplex Trailer Configuration can sweep 300000 m^2/hour *)
nRunwaySweeps = CrackGrowthModel[[11, 2]]; (* Number of runway sweeps per operational day *)
TvisualInspection = 0.5; (* Length of visual inspection in hours *)
InspectionVehicleSpeed = 30; (* Given in mph *)
POD = 0.50;
Naircraft = 1200; (* Number of Aircraft that may use runway *)
Wsweeper = 7.32; (* Sweeper width *)
ScanRate = CrackGrowthModel[[11, 1]];
(* Automated Inspections System Scan Rate, 1 scan every 3 hours *)
Tremoval = 0.5;
(* Time to remove/retrieve FOD detected by Automated FOD detection systems *)
ηscan = CrackGrowthModel[[11, 3]];
ηremoval = 0.9;
NIntervalInspections = 2; (* Two  Engine interval inspections,
first at 1/2 90th percentile comp. life, second at 3/4 90th percentile comp. life *)
EngineIntervalInspections = CrackGrowthModel[[11, 5]];
(* Engine Non-Destructive Inspections on (1) or off (0) *)


R = 0.80; (* Load ratio σmin/σmax *)
StressFrequency = 25; (* Load frequency, spectrum *)
Nincrement = 2 * 60 * 60 * StressFrequency; (* Number of load cycles per takeoff *)
aaD = 0.000381;
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
(* Minimum detectable flaw size in meters with current Non Destructive Inspection methods *)
FodDiam = 0.00133;
Ntakeoffs = 300; (* Number of takeoffs per engine Life Time,
takeoffs before 1st major teardown removal/overhaul of engine *)
NcyclesIntervalInspection = 27 000 000;
(* 1/2 90th percentile cycles for Life Time of 300 takeoffs *)


(* Life Cycle Costing, Inspection Labor Costs *)

FLTS = 135; (* Flights per year per Aircraft *)
NA = 1200; (* Number of Aircraft, set to 1200 for Fleet *)
QTY = 26; (* Component quantity per Aircraft, 2 engines with 13 rotor disks *)
AVEFLTHR = 2; (* Average flight hour per flight *)
BurdenedDIRLAB = 111 426.95 / 2080;
(* Burdened labor cost per hour, in 2015 US $ *)
SCHMNTINSMM = 8;
(* Scheduled Inspection Labor for 1/2 90th percentile comp. life
 and at at 3/4 90th percentile comp. life, for 2 personnel at 4 hours each *)
RECMHRSrepairs = 16;
(* Rectification Man Hours for 1/2 90th percentile
 comp. life  and at at 3/4 90th percentile comp. life, repairs,
for four personnel at 4 hours each, partial engine removal, access pannels open *)
RECMHRSreplacements = 8;
(* Rectification Man Hours for 1/2 90th percentile
 comp. life  and at at 3/4 90th percentile comp. life, replacements,
for four personnel at 2 hours each, engine is not removed, only access pannels *)
SCHCAMArepairs = 1000; (* Scheduled Corrective Action Material *)
CAMAreplacements = 5000;
SCHINCOST = EngineIntervalInspections *
  NIntervalInspections * (FLTS * NA * QTY * AVEFLTHR * SCHMNTINSMM * BurdenedDIRLAB) /
   (NcyclesIntervalInspection / (60 * 60 * StressFrequency));
(* Burdened Labor Cost per Scheduled Inspection *)
Print["Burdened Labor Cost per Scheduled Inspection of Engine: ", SCHINCOST];
MHRSEnginereplacements = 3; (* Man Hours for an Engine Replacement *)
CAMAEnginereplacements = 2 460 000; (* Cost of Engine, 2015 US$ *)

MilitaryThrust = 79.1; (* kN *)
SpecificFuelConsumption = 77.5; (* kg/(kN·h) at MilitaryThrust *)
CurrentPriceofJetFuel = 1.02; (* Price per gallon, IATA current price *)
CostofFuelperFlight = MilitaryThrust * SpecificFuelConsumption * 2 * 0.3261 * CurrentPriceofJetFuel;
(* two hour flight,  0.3261 is conversion from kg to gallons of fuel *)


CostofAcquisitionFODSweeper = 3 * 7000;
(* Sweeper data is for the FOD*Boss in the Triplex Trailer Configuration *)
CostofVehicleUseperMile = 0.1375;
(* Operating costs of light truck per mile, 2014 data, 2015 is not available yet *)


LSPTreatment = CrackGrowthModel[[11, 4]];
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
4 |
(* 2 if LSP Treatment is considered or 0 if not *)

CostofAcquisitionLSPMachine = 66976 + RandomReal[] * (109702 - 66976);
(* LSP cost data from Shukla's paper *)
LSPProcessTime = 1 + RandomReal[] * 9;
(* Process time in minutes, LSP data from Shukla's paper *)
LSPCleanUpTime = 5; (* Cleanup time in minutes, LSP data from Shukla's paper *)
LSPChageOverTime = 10;
(* Change over time between LSP processes in minutes, LSP data from Shukla's paper *)
LSPProcessCost = NA * QTY * (nrotorblades *
    (LSPTreatment * LSPProcessTime + LSPCleanUpTime + LSPChageOverTime) / 60 * BurdenedDIRLAB);

Do[
 Print[Style["Takeoff Lifetime: ", 36, Red], n];
 Ncycles = 0;
 NcyclesNaturalGrowth = 0;
 NcylesAppended = {};
 NcyclesAppendedTotal = {};
 NcyclesNaturalGrowthAppended = {};
 aaAppended = {};
 aaAppendedNaturalGrowth = {};
 Ncyclesbladereplaced = 0;
 Nbladereplacements = 0;
 NcyclesatImpact = {};
 CostIncreaseDuetoEngineEfficiencyDregradation = 0;

 LifeTimerepairs = Nrepairs;
 LifeTimereplacements = Nreplacements;
 LifeTimeEnginereplacements = NEnginereplacements;

 RandomNumberforAutomatedInspection = RandomReal[];
 CostofAcquisitionAutomatedFODDetectionSystem = (17 / 0.09 * ηscan - 121.777777778) * 10^6;
 (* AutomatedFODDetectionSystem is for several systems *)
 TotalCostofAutomatedInspectionsandRunwaySweep = 0;
 TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisition = 0;
 Do[TotalCostofAutomatedInspectionsandRunwaySweep =
   2.704 * ((nRunwaySweeps * (Lrunway * 0.000621371) * (Wrunway / Wsweeper) * CostofVehicleUseperMile) +
       ((Lrunway * Wrunway / FODBossSweptAreaperhour) * nRunwaySweeps * 2 * BurdenedDIRLAB) +
       (24 * ScanRate) * Tremoval * BurdenedDIRLAB) +
     TotalCostofAutomatedInspectionsandRunwaySweep, {nn, 1, Ntakeoffs}];
 TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisition =
  CostofAcquisitionFODSweeper / Naircraft + CostofAcquisitionAutomatedFODDetectionSystem / Naircraft +
   TotalCostofAutomatedInspectionsandRunwaySweep +
   CostofAcquisitionLSPMachine / Naircraft + LSPProcessCost;
 Print["Total Cost of Automated Inspections and Runway Sweep
    and Sweeper Acquisition per takeoff Life Time: ",
  TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisition];

 Print["TotalCostofAutomatedInspectionsandRunwaySweep: ",
  TotalCostofAutomatedInspectionsandRunwaySweep];
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
TotalCostofAutomatedInspectionsandRunwaySweep = 0;

Print["CostofAcquisitionAutomatedFODDetectionSystem/Aircraft: ",
 CostofAcquisitionAutomatedFODDetectionSystem / Aircraft];

(* Probability of FOD under/within Vortex Core from Automated Inspections *)
Do[
 Label[Restart];

 PoFodunderVortexCore =
  (1/Lrunway) (-0.25` Lrunway + Wrunway (-0.43301270189221924` + 0.43301270189221924`
        Sqrt[(1.` + 1/Wrunway^2 Lrunway (-3.4641016151377553` Wrunway + Lrunway
          (0.3333333333333334` + Wrunway^2 (4.618802153517007` - 0.19245008972987526 *
              ((nRunwaySweeps * Wrunway * Lrunway * ηsweep / FODBossSweptAreaperhour +
                 (24 * ScanRate) (1 / ScanRate - Tremoval) ηscan ηremoval))) *
            ρFodonRunway))]]) * (Dstagpoint - DParticle / 2) / Wrunway;
 ProbAspiration = PoFodunderVortexCore;

 RandomNumber = RandomReal[];
 (*Print["RandomNumber: ", RandomNumber];*)

 If[RandomNumber ≤ ProbAspiration,
  Print["RandomNumber that Triggered Particle Aspiration to Fan Face: ", RandomNumber];

  PoRotorStrike = 402 / 1081;

  If[RandomReal[] ≤ PoRotorStrike,

   RandomNumberforInEngineParticleMotion = RandomReal[];
   h = (0.33 - 0.24) * RandomReal[] + 0.24;
   GroundDistance = (1 - 0.5) * RandomReal[] + 0.5;
   InletMachNumber = 0.45 + 0.05 * RandomReal[];
   HeadWind = (12 - 9.357) * RandomReal[] + 9.357;

   Axialσ = (1 063 168 750.67919 + 515 059 837.719837 * h + -366 022.249681334 * GroundDistance +
       -2 123 990 818.12455 * InletMachNumber + -5 893 409.84568672 * HeadWind +
       (h - 0.285513648014925) * ((h - 0.285513648014925) * -115 914 736.246555) +
       (h - 0.285513648014925) * ((GroundDistance - 0.727088406885572) * -3 860 239.90929367) +
       (GroundDistance - 0.727088406885572) *
        ((GroundDistance - 0.727088406885572) * 601 596.125270522) +
       (h - 0.285513648014925) * ((InletMachNumber - 0.472574813666666) * -1 770 961 724.2888) +
       (GroundDistance - 0.727088406885572) * ((InletMachNumber - 0.472574813666666) *
          -3 601 984.15779136) + (InletMachNumber - 0.472574813666666) *
        ((InletMachNumber - 0.472574813666666) * -1 852 995 080.65505) +
       (h - 0.285513648014925) * ((HeadWind - 10.6542844452363) * 1 815 221.56210402) +
       (GroundDistance - 0.727088406885572) * ((HeadWind - 10.6542844452363) * -25 325.6256442044) +
       (InletMachNumber - 0.472574813666666) * ((HeadWind - 10.6542844452363) * -12 296 059.8523198) +
       (HeadWind - 10.6542844452363) *
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
6 |
          ((HeadWind - 10.6542844452363) * 222250.495346287)) * (10500 / 14000) ^ 2;

  ChordatImpact = 0.0233114081179261 + -0.00020975222191373 3 * h + 0.000000177621436898 9 *
      GroundDistance + 0.000862890506036301 * InletMachNumber + 0.0000023869436839637 * HeadWind +
     (h - 0.285513648014925) * ((h - 0.285513648014925) * 0.000002167677195839) +
     (h - 0.285513648014925) * ((GroundDistance - 0.727088406885572) * 0.000000271231340696 2) +
     (GroundDistance - 0.727088406885572) *
      ((GroundDistance - 0.727088406885572) * -0.000000484901468221 3) +
     (h - 0.285513648014925) * ((InletMachNumber - 0.472574813666666) * 0.00106043173745756) +
     (GroundDistance - 0.727088406885572) * ((InletMachNumber - 0.472574813666666) *
        0.0000017569184995802) + (InletMachNumber - 0.472574813666666) *
      ((InletMachNumber - 0.472574813666666) * -0.000009744268109273 5) +
     (h - 0.285513648014925) * ((HeadWind - 10.6542844452363) * -0.000000041774235168 7) +
     (GroundDistance - 0.727088406885572) * ((HeadWind - 10.6542844452363) * -0.000000004437754409 6) +
     (InletMachNumber - 0.472574813666666) *
      ((HeadWind - 10.6542844452363) * -0.000000067866720 7) +
     (HeadWind - 10.6542844452363) * ((HeadWind - 10.6542844452363) * -0.000000080066712265 5);

  Rimpact = 0.065415445705254 + -0.156714564501262 * h + 0.000132668082703971 * GroundDistance +
     0.64470128769476 * InletMachNumber + 0.00178339145613628 * HeadWind +
     (h - 0.285513648014925) * ((h - 0.285513648014925) * 0.00161247997422258) +
     (h - 0.285513648014925) * ((GroundDistance - 0.727088406885572) * 0.000201935239468427) +
     (GroundDistance - 0.727088406885572) *
      ((GroundDistance - 0.727088406885572) * -0.000362165459376822) +
     (h - 0.285513648014925) * ((InletMachNumber - 0.472574813666666) * 0.792303049785879) +
     (GroundDistance - 0.727088406885572) * ((InletMachNumber - 0.472574813666666) *
        0.00131731731841904) + (InletMachNumber - 0.472574813666666) *
      ((InletMachNumber - 0.472574813666666) * -0.007329660244592 97) +
     (h - 0.285513648014925) * ((HeadWind - 10.6542844452363) * -0.000031206878645552) +
     (GroundDistance - 0.727088406885572) * ((HeadWind - 10.6542844452363) * -0.000003242570130508) +
     (InletMachNumber - 0.472574813666666) *
      ((HeadWind - 10.6542844452363) * -0.000051684121385094 1) +
     (HeadWind - 10.6542844452363) * ((HeadWind - 10.6542844452363) * -0.000059816864478025 7);

  XpercentChordmin = 0;
XpercentChordmax = 0.94719;
  alphaXpercentChord = 0.8478;
  betaXpercentChord = 1.59718;

  XpercentChord = InverseBetaRegularized[RandomReal[], alphaXpercentChord, betaXpercentChord] *
      (XpercentChordmax - XpercentChordmin) + XpercentChordmin;

  DepthofPenetrationmin = 0.000029;
DepthofPenetrationmax = 0.00051;
  alphaDepthofPenetration = 1.4797852;
  betaDepthofPenetration = 3.5553198;

  DepthofPenetration =
   InverseBetaRegularized[RandomReal[], alphaDepthofPenetration, betaDepthofPenetration] *
     (DepthofPenetrationmax - DepthofPenetrationmin) + DepthofPenetrationmin;

Printed by Wolfram Mathematica Student Edition
```

Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
LocalThicknessatImpact =
 0.00186625343042413 + 0.00132154137591184 * XpercentChord + (XpercentChord - 0.317678693072139) *
    ((XpercentChord - 0.317678693072139) * -0.0080516775782884) +
   (XpercentChord - 0.317678693072139) * ((XpercentChord - 0.317678693072139) *
      ((XpercentChord - 0.317678693072139) * 0.00548564206578457)) +
   (XpercentChord - 0.317678693072139) * ((XpercentChord - 0.317678693072139) * ((XpercentChord -
         0.317678693072139) * ((XpercentChord - 0.317678693072139) * -0.0402999953775497))) +
   (XpercentChord - 0.317678693072139) * ((XpercentChord - 0.317678693072139) *
      ((XpercentChord - 0.317678693072139) * ((XpercentChord - 0.317678693072139) *
         ((XpercentChord - 0.317678693072139) * 0.0530674615909787))));

ImpactSide = If[RandomReal[] ≤ 283 / 402, "Pressure Side", "Suction Side"];

EIFSa =0.00025106738269976316;
EIFsDiam = 2 * EIFSa;
aa = EIFSa;
Diameter2c = EIFsDiam;
aa0 = EIFSa;
aaNaturalGrowth = EIFSa;
Diameter2cNaturalGrowth = 2 * EIFSa;

BetaCompoundedAppend = {};
BetaCompoundedAppendNaturalGrowth = {};
Impact1 = 0;
ChordatImpact0 = ChordatImpact;
XpercentChord0 = XpercentChord; (*XpercentChord;*)
DeltaSigma = 2 * Axialσ * (1 - R) / (1 + R);
ImpactEccentricty = ChordatImpact0 * Abs[XpercentChord0 - 0.5];
Print["ImpactEccentricty: ", ImpactEccentricty],
Axialσ = 0;
ChordatImpact = 0;
XpercentChord = 0;
DepthofPenetration = 0;
LocalThicknessatImpact = 0;
];

If[ChordatImpact == 0, i = i + 1;
 If[i == Ntakeoffs, II = 1;
  Print["No Particle Impacts During Life of Blade"];
  Goto[End2], Goto[Restart]], If[i == Ntakeoffs, Goto[End1]]];

If[0.05 ≤ XpercentChord ≤ 0.80, AverageAirfoilThickness = LocalThicknessatImpact,
 AverageAirfoilThickness = 0.0551 * ChordatImpact0];
MaxAirfoilThickness = 0.10 * ChordatImpact;
Kic = 222.794957946974 +
  -9579.98234775552 * MaxAirfoilThickness + (MaxAirfoilThickness - 0.0079022222222222) *
    ((MaxAirfoilThickness - 0.0079022222222222) * 164325.655251741) +
   (MaxAirfoilThickness - 0.0079022222222222) * ((MaxAirfoilThickness - 0.0079022222222222) *
      ((MaxAirfoilThickness - 0.0079022222222222) * 89110035.2890032)) +
   (MaxAirfoilThickness - 0.0079022222222222) * ((MaxAirfoilThickness - 0.0079022222222222) *
      ((MaxAirfoilThickness - 0.0079022222222222) *
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
8 |
          ((MaxAirfoilThickness - 0.00790222222222222) * -34579079833.1077))) +
     (MaxAirfoilThickness - 0.00790222222222222) * ((MaxAirfoilThickness - 0.00790222222222222) *
        ((MaxAirfoilThickness - 0.00790222222222222) * ((MaxAirfoilThickness - 0.00790222222222222) *
           ((MaxAirfoilThickness - 0.00790222222222222) * 2523050943520.48))));

If[ChordatImpact > 0,
 Impact1 = 1;
 Print["Takeoff When Impact Occurs: ", Impact1];
 Print["XpercentChord: ", XpercentChord];

 Nimpacts = Nimpacts + 1;

 AxialσMax = 2 * Axialσ / (1 + R);
 KMaxatFracture = K1c;

 If[(0.80 ≥ XpercentChord ≥ 0.05) && ChordatImpact > 0,

  aac50thpercentilethumbnailflaw = 0.000650084093009177;
  aac = aac50thpercentilethumbnailflaw;

  (*Clear[aac];
  aac=aac/.Extract[
     FindMinimum[{{{1.12*(1.08882991825178+0.191072321954091*(aac/AverageAirfoilThickness)+
               -0.2275*(aac/(2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                    ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5]))+
               ((aac/AverageAirfoilThickness)-0.2375)*(((aac/AverageAirfoilThickness)-
                    0.2375)*-0.336251716696097)+((aac/AverageAirfoilThickness)-0.2375)*
                ((((aac/(2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                          ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^
                           0.5]))-0.256666666666667)*-0.642299410757611)+
               ((aac/(2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                        ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5]))-0.256666666666667)*
                ((((aac/(2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                          ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5]))-0.256666666666667)*
                  0.443509615384615))*(P1*(Log[(ChordatImpact/2+ImpactEccentricty+
                     ((ChordatImpact/2+ImpactEccentricty)^2-
                        ((2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                             ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)^2)^0.5)/
                   ((2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                        ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)])/
                ((ArcCos[(2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                          ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/
                      (ChordatImpact-2*ImpactEccentricty)])*(Log[(ChordatImpact/2+
                        ImpactEccentricty+((ChordatImpact/2+ImpactEccentricty)^2-
                           ((2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                                ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)^2)^0.5)/
                      ((2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                           ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)])+
                  (ArcCos[(2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^
                             0.5, ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/
                      (ChordatImpact+2*ImpactEccentricty)])*(Log[(ChordatImpact/2-
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
                ImpactEccentricty+((ChordatImpact/2-ImpactEccentricty)^2-
                    ((2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                        ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)^2)^0.5)/
                  ((2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                    ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)])))*
              (((aac/((2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                  ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2))^0.5)/
                (1/4 π (1+(aac)^2/If[2 aac≤FodDiam, (aac (-aac+FodDiam))^0.5`, 0.5`
                    ((FodDiam)^2)^0.5`]^2)))*(P1*aac)^0.5)-
              (KMaxatFracture)/((Axial$\sigma$/10^6)))^2, aac≥0}, {aac, 0.0001)], {2}];*)


(*Nf=Quiet[NIntegrate[1/(0.0000000000052)*
    1/(((1.12*(1.08882991825178+0.191072321954091*(aa/AverageAirfoilThickness)+
          -0.2275*(aa/(2*If[aa≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aa)^2)^0.5,
              ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5]))+
          ((aa/AverageAirfoilThickness)-0.2375)*(((aa/AverageAirfoilThickness)-
              0.2375)*-0.336251716696097)+((aa/AverageAirfoilThickness)-0.2375)*
           (((aa/(2*If[aa≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aa)^2)^0.5,
                ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^
                 0.5]))-0.256666666666667)*-0.642299410757611)+
          ((aa/(2*If[aa≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aa)^2)^0.5,
              ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5]))-0.256666666666667)*
           (((aa/(2*If[aa≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aa)^2)^0.5,
                ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5]))-0.256666666666667)*
            0.443509615384615))*(P1*(Log[(ChordatImpact/2+ImpactEccentricty+
              ((ChordatImpact/2+ImpactEccentricty)^2-
                ((2*If[aa≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aa)^2)^0.5,
                    ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)^2)^0.5)/
             ((2*If[aa≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aa)^2)^0.5,
                ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)])/
           ((ArcCos[(2*If[aa≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aa)^2)^0.5,
                ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/
              (ChordatImpact-2*ImpactEccentricty)])*(Log[(ChordatImpact/2+
               ImpactEccentricty+((ChordatImpact/2+ImpactEccentricty)^2-
                  ((2*If[aa≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aa)^2)^0.5,
                      ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)^2)^0.5)/
                ((2*If[aa≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aa)^2)^0.5,
                    ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)])+
             (ArcCos[(2*If[aa≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aa)^2)^
                  0.5, ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/
                (ChordatImpact+2*ImpactEccentricty)])*(Log[(ChordatImpact/2-
                 ImpactEccentricty+((ChordatImpact/2-ImpactEccentricty)^2-
                    ((2*If[aa≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aa)^2)^0.5,
                        ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)^2)^0.5)/
                  ((2*If[aa≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aa)^2)^0.5,
                      ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)])))*
        (((aa/((2*If[aa≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aa)^2)^0.5,
            ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2))^0.5)/NIntegrate[
          (1-(1-(aa/((2*If[aa≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aa)^2)^
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
10 |                                                    0.5, ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/
                                                2))^2)*(Sin[φ])^2), {φ, 0, P1/2}])*(DeltaSigma/10^6)*
                                  (P1*aa)^0.5)^2.5)*(KMaxatFracture)^0.67), (aa, aa0, aac)]]; *)

Print["aac: ", aac];

Do[
 aaAppendedNaturalGrowth = AppendTo[aaAppendedNaturalGrowth, aaNaturalGrowth];
 NcyclesNaturalGrowthAppended =
  AppendTo[NcyclesNaturalGrowthAppended, NcyclesNaturalGrowth];
 BetaCFC = ((aaNaturalGrowth / (Diameter2cNaturalGrowth / 2))^0.5) / NIntegrate[
    (1 - (1 - (aaNaturalGrowth / (Diameter2cNaturalGrowth / 2))^2) * (Sin[φ])^2), {φ, 0, P1 / 2}];
 BetaBFS = 1.12;
 G1 = ArcCos[Diameter2cNaturalGrowth / (ChordatImpact - 2 * ImpactEccentricty)];
 G2 = ArcCos[Diameter2cNaturalGrowth / (ChordatImpact + 2 * ImpactEccentricty)];
 G3 = Log[(ChordatImpact / 2 - ImpactEccentricty + ((ChordatImpact / 2 - ImpactEccentricty)^
          2 - (Diameter2cNaturalGrowth / 2)^2)^0.5) / (Diameter2cNaturalGrowth / 2)];
 G4 = Log[(ChordatImpact / 2 + ImpactEccentricty + ((ChordatImpact / 2 + ImpactEccentricty)^
          2 - (Diameter2cNaturalGrowth / 2)^2)^0.5) / (Diameter2cNaturalGrowth / 2)];
 BetaW = P1 * G4 / (G1 * G4 + G2 * G3);
 BetaFFS = 1.08882991825178 + 0.191072321954091 * (aaNaturalGrowth / AverageAirfoilThickness) +
   -0.2275 * (aaNaturalGrowth / Diameter2cNaturalGrowth) +
   ((aaNaturalGrowth / AverageAirfoilThickness) - 0.2375) *
    (((aaNaturalGrowth / AverageAirfoilThickness) - 0.2375) * -0.336251716696097) +
   ((aaNaturalGrowth / AverageAirfoilThickness) - 0.2375) *
    (((aaNaturalGrowth / Diameter2cNaturalGrowth) - 0.256666666666667) * -0.642299410757611) +
   ((aaNaturalGrowth / Diameter2cNaturalGrowth) - 0.256666666666667) *
    (((aaNaturalGrowth / Diameter2cNaturalGrowth) - 0.256666666666667) * 0.443509615384615);
 BetaCompounded = BetaBFS * BetaFFS * BetaW * BetaCFC;
 BetaCompoundedAppendNaturalGrowth =
  AppendTo[BetaCompoundedAppendNaturalGrowth, BetaCompounded];
 DeltaK = BetaCompounded * DeltaSigma * (P1 * aaNaturalGrowth)^0.5;
 σResidual = 0;
 If[ImpactSide == "Pressure Side",
  If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 1, σResidual = -InverseCDF[
        NormalDistribution[Abs[(-70.7331771213364) + 117 116.263407782 * aaNaturalGrowth +
            (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * -395 474 551.758495 +
            (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth -
                0.000335) * 600 608 204 212.839 + (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth -
                0.000335) * (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
             -313 392 042 189 632], Abs[((-91.388930233299) + 163 881.618156941 * aaNaturalGrowth +
              (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * -621 710 641.059832 +
              (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
               (aaNaturalGrowth - 0.000335) * 784 096 231 132.582 + (aaNaturalGrowth - 0.000335) *
               (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
               (aaNaturalGrowth - 0.000335) * -181 928 735 061 177) - ((-51.0322827764947) +
              73 621.0917592117 * aaNaturalGrowth + (aaNaturalGrowth - 0.000335) *
               (aaNaturalGrowth - 0.000335) * -186 084 270.781473 + (aaNaturalGrowth - 0.000335) *
               (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * 369 136 849 298.813 +
              (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth -
                  0.000335) * (aaNaturalGrowth - 0.000335) * -331 072 585 461 178)] / 6],
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
        RandomReal[]] * If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1,
        If[LSPTreatment == 3, 1.25, If[LSPTreatment == 0, 0]]]]],
  If[ImpactSide == "Suction Side", If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 0.80,
     σResidual = -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) +
           117 116.263407782 * aaNaturalGrowth + (aaNaturalGrowth - 0.000335) *
            (aaNaturalGrowth - 0.000335) * -395 474 551.758495 + (aaNaturalGrowth - 0.000335) *
            (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * 600 608 204 212.839 +
           (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
            (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * -313 392 042 189 632],
         Abs[((-91.388930233299) + 163 881.618156941 * aaNaturalGrowth +
              (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * -621 710 641.059832 +
              (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
               (aaNaturalGrowth - 0.000335) * 784 096 231 132.582 +
              (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth -
                 0.000335) * (aaNaturalGrowth - 0.000335) * -181 928 735 061 177) -
            ((-51.0322827764947) + 73 621.0917592117 * aaNaturalGrowth +
              (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * -186 084 270.781473 +
              (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
               (aaNaturalGrowth - 0.000335) * 369 136 849 298.813 + (aaNaturalGrowth -
                 0.000335) * (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
               (aaNaturalGrowth - 0.000335) * -331 072 585 461 178)] / 6], RandomReal[]] *
       If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1, If[LSPTreatment == 3,
          1.25, If[LSPTreatment == 0, 0]]]]]]]];

 Kmax = BetaCompounded * If[aaNaturalGrowth ≤ 0.00091,
    AxialσMax + σResidual * 10^6, AxialσMax] * (P1 * aaNaturalGrowth)^0.5;
 dadN = (0.0000000000052) * ((DeltaK / 10^6)^2.5) * ((Kmax / 10^6)^0.67);
 DeltaNcycles = Nincrement;
 daa = dadN * DeltaNcycles;
 aaNaturalGrowth = aaNaturalGrowth + daa;
 If[aaNaturalGrowth === ComplexInfinity || Head[aaNaturalGrowth] === Complex,
  aaNaturalGrowth = Re[aaNaturalGrowth]];
 Diameter2cNaturalGrowth = 2 * aaNaturalGrowth;
 NcyclesNaturalGrowth = 11 * Nincrement;
 If[aaNaturalGrowth ≥ aac || Ntakeoffs * 2 * 60 * 60 * StressFrequency == Ncycles,
  Print["ith takeoff during Natural Growth when a >= aac or
      Ncycles = Ntakeoffs*2*60*60*StressFrequency : ", ii];
  NcyclesNaturalGrowth = 0; Goto[Next2]],
 {ii, 1, Ntakeoffs}];

Label[Next2];


Do[
 aaAppended = AppendTo[aaAppended, aa];
 NcylesAppended = AppendTo[NcylesAppended, Ncycles];
 BetaCFC = ((aa / (Diameter2c / 2))^0.5) /
   NIntegrate[(1 - (1 - (aa / (Diameter2c / 2)))^2) * (Sin[φ])^2), {φ, 0, P1 / 2}];
 BetaBFS = 1.12;
 G1 = ArcCos[Diameter2c / (ChordatImpact - 2 * ImpactEccentricty)];
 G2 = ArcCos[Diameter2c / (ChordatImpact + 2 * ImpactEccentricty)];
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
12 |
G3 = Log[(ChordatImpact / 2 - ImpactEccentricty +
      ((ChordatImpact / 2 - ImpactEccentricty)^2 - (Diameter2c / 2)^2)^0.5) / (Diameter2c / 2)];
G4 = Log[(ChordatImpact / 2 + ImpactEccentricty + ((ChordatImpact / 2 + ImpactEccentricty)^
         2 - (Diameter2c / 2)^2)^0.5) / (Diameter2c / 2)];
BetaW = P1 * G4 / (G1 * G4 + G2 * G3);
BetaFFS = 1.08882991825178 + 0.191072321954091 * (aa / AverageAirfoilThickness) +
  -0.2275 * (aa / Diameter2c) + ((aa / AverageAirfoilThickness) - 0.2375) *
   (((aa / AverageAirfoilThickness) - 0.2375) * -0.336251716696097) +
  ((aa / AverageAirfoilThickness) - 0.2375) * (((aa / Diameter2c) - 0.256666666666667) *
     -0.642299410757611) + ((aa / Diameter2c) - 0.256666666666667) *
   (((aa / Diameter2c) - 0.256666666666667) * 0.443509615384615);
BetaCompounded = BetaBFS * BetaFFS * BetaW * BetaCFC;
BetaCompoundedAppend = AppendTo[BetaCompoundedAppend, BetaCompounded];
DeltaK = BetaCompounded * DeltaSigma * (P1 * aa)^0.5;
σResidual = 0;
If[ImpactSide == "Pressure Side", If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 1,
   σResidual = -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) +
         117 116.263407782 * aa + (aa - 0.000335) * (aa - 0.000335) * -395 474 551.758495 +
         (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * 600 608 204 212.839 + (aa - 0.000335) *
          (aa - 0.000335) * (aa - 0.000335) * -313 392 042 189 632],
        Abs[((-91.388930233299) + 163 881.618156941 * aa + (aa - 0.000335) * (aa - 0.000335) *
            -621 710 641.059832 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
           784 096 231 132.582 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
           (aa - 0.000335) * -181 928 735 061 177) - ((-51.0322827764947) + 73 621.0917592117 *
           aa + (aa - 0.000335) * (aa - 0.000335) * -186 084 270.781473 + (aa - 0.000335) *
           (aa - 0.000335) * (aa - 0.000335) * 369 136 849 298.813 + (aa - 0.000335) *
           (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -331 072 585 461 178)] / 6],
      RandomReal[] * If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1,
      If[LSPTreatment == 3, 1.25, If[LSPTreatment == 0, 0]]]]],
 If[ImpactSide == "Suction Side", If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 0.80,
   σResidual = -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) + 117 116.263407782 *
         aa + (aa - 0.000335) * (aa - 0.000335) * -395 474 551.758495 + (aa - 0.000335) *
         (aa - 0.000335) * (aa - 0.000335) * 600 608 204 212.839 + (aa - 0.000335) *
         (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -313 392 042 189 632],
        Abs[((-91.388930233299) + 163 881.618156941 * aa + (aa - 0.000335) * (aa - 0.000335) *
            -621 710 641.059832 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
           784 096 231 132.582 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
           (aa - 0.000335) * -181 928 735 061 177) - ((-51.0322827764947) + 73 621.0917592117 *
           aa + (aa - 0.000335) * (aa - 0.000335) * -186 084 270.781473 + (aa - 0.000335) *
           (aa - 0.000335) * (aa - 0.000335) * 369 136 849 298.813 + (aa - 0.000335) *
           (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -331 072 585 461 178)] / 6],
      RandomReal[] * If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1, If[
       LSPTreatment == 3, 1.25, If[LSPTreatment == 0, 0]]]]]];
Kmax = BetaCompounded * If[aa ≤ 0.00091, AxialσMax + σResidual * 10^6, AxialσMax] * (P1 * aa)^0.5;
dadN = (0.0000000000052) * ((DeltaK / 10^6)^2.5) * ((Kmax / 10^6)^0.67);
DeltaNcycles = Nincrement;
daa = dadN * DeltaNcycles;
aa = aa + daa;
If[aa === ComplexInfinity || Head[aa] === Complex, aa = Re[aa]];
Diameter2c = 2 * aa;
Ncycles = Ncycles + Nincrement;
```

Printed by Wolfram Mathematica Student Edition

Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
LevelIIIDamageRandomSeed = RandomReal[];
If[EngineIntervalInspections == 1,
 If[Ncycles == NcyclesIntervalInspection && aaD ≤ aa < aac, Nrepairs = Nrepairs + 1;
  aa = EIFSa, If[LevelIIIDamageRandomSeed ≥ 4 * 10^-5,
   If[Ncycles == NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
    aa = EIFSa], If[LevelIIIDamageRandomSeed < 4 * 10^-5 && Ncycles ==
      NcyclesIntervalInspection && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
 If[LevelIIIDamageRandomSeed > 4 * 10^-5,
  If[Ncycles < NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
   aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles <
      NcyclesIntervalInspection && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
 If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[NcyclesIntervalInspection * 1.5 > Ncycles >
     NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
   aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5,
   If[NcyclesIntervalInspection * 1.5 > Ncycles > NcyclesIntervalInspection && aa ≥ aac,
    NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
 If[Ncycles == NcyclesIntervalInspection * 1.5 && aaD ≤ aa < aac, Nrepairs = Nrepairs + 1;
  aa = EIFSa, If[LevelIIIDamageRandomSeed ≥ 4 * 10^-5,
   If[Ncycles == NcyclesIntervalInspection * 1.5 && aa ≥ aac, Nreplacements = Nreplacements + 1;
    aa = EIFSa], If[LevelIIIDamageRandomSeed < 4 * 10^-5 && Ncycles ==
      NcyclesIntervalInspection * 1.5 && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
 If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[Ntakeoffs * 2 * 60 * 60 * StressFrequency > Ncycles >
     NcyclesIntervalInspection * 1.5 && aa ≥ aac, Nreplacements = Nreplacements + 1;
   aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles >
      NcyclesIntervalInspection * 1.5 && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]],
 If[EngineIntervalInspections == 0, If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[Ncycles <
      Ntakeoffs * 2 * 60 * 60 * StressFrequency && aa ≥ aac, Nreplacements = Nreplacements + 1;
    aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles < Ntakeoffs * 2 *
        60 * 60 * StressFrequency && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
     aa = EIFSa;
     Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]]],
 {i1, 1, Impact1}];

Label[Next1];

Print["Nrepairs: ", Nrepairs];
Print["Nreplacements: ", Nreplacements];
Print["NEngineReplacements: ", NEngineReplacements];

If[aa ≤ DepthofPenetration, aa = DepthofPenetration;
 Diameter2c = 2 * If[aa ≤ FodDiam/2, ((FodDiam/2)^2 - (FodDiam/2 - aa)^2)^0.5,
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
14 |
        ((FodDiam/2)^2 - (FodDiam/2 - FodDiam/2)^2)^0.5], Diameter2c = 2 * aa];

(*aa0regressed=aa;
Do[
 BetaCFC=((aa0regressed/(Diameter2c/2))^0.5)/
   NIntegrate[(1-(1-(aa0regressed/(Diameter2c/2))^2)*(Sin[φ])^2),{φ,0,P1/2}];
 BetaBFS=1.12;
 G1=ArcCos[Diameter2c/(ChordatImpact-2*ImpactEccentricty)];
 G2=ArcCos[Diameter2c/(ChordatImpact+2*ImpactEccentricty)];
 G3=Log[(ChordatImpact/2-ImpactEccentricty+
       ((ChordatImpact/2-ImpactEccentricty)^2-(Diameter2c/2)^2)^0.5)/(Diameter2c/2)];
 G4=Log[(ChordatImpact/2+ImpactEccentricty+((ChordatImpact/2+ImpactEccentricty)^
          2-(Diameter2c/2)^2)^0.5)/(Diameter2c/2)];
 BetaW=P1*G4/(G1*G4+G2*G3);
 BetaFFS=1.08882991825178+0.191072321954091*(aa0regressed/AverageAirfoilThickness)+
   -0.2275*(aa0regressed/Diameter2c)+((aa0regressed/AverageAirfoilThickness)-0.2375)*
    (((aa0regressed/AverageAirfoilThickness)-0.2375)*-0.336251716696097)+
   ((aa0regressed/AverageAirfoilThickness)-0.2375)*
    (((aa0regressed/Diameter2c)-0.256666666666667)*-0.642299410757611)+
   ((aa0regressed/Diameter2c)-0.256666666666667)*
    (((aa0regressed/Diameter2c)-0.256666666666667)*0.443509615384615);
 BetaCompounded=BetaBFS*BetaFFS*BetaW*BetaCFC;
 BetaCompoundedAppend=AppendTo[BetaCompoundedAppend,BetaCompounded];
 DeltaK=BetaCompounded*DeltaSigma*(P1*aa0regressed)^0.5;
 Kmax=BetaCompounded*AxialcMax*(P1*aa0regressed)^0.5;
 aa0regressed=
  aa0regressed-(0.0000000000052)*((DeltaK/10^6)^2.5)*((Kmax/10^6)^0.67)*Nincrement,
 (ii,1,Impacti+1)];


Print["aa0regressed: ",aa0regressed," aa at Impact: ",aa];
If[aa0regressed===ComplexInfinity||Head[aa0regressed]===Complex,
 Goto[Next3],aa0regressedAppended=AppendTo[aa0regressedAppended,aa0regressed]];
Label[Next3];*)


aaAppended = AppendTo[aaAppended, aa];
NcyclesAppended = AppendTo[NcyclesAppended, Ncycles];
Print["Ncycles Between Initial Growth and Impact: ", Ncycles];
Print["NcyclesatImpact: ", Impacti * Nincrement];

Do[
 BetaCFC = ((aa/(Diameter2c/2))^0.5)/
   NIntegrate[(1-(1-(aa/(Diameter2c/2))^2)*(Sin[φ])^2), {φ, 0, P1/2}];
 BetaBFS = 1.12;
 G1 = ArcCos[Diameter2c/(ChordatImpact-2*ImpactEccentricty)];
 G2 = ArcCos[Diameter2c/(ChordatImpact+2*ImpactEccentricty)];
 G3 = Log[(ChordatImpact/2-ImpactEccentricty+
       ((ChordatImpact/2-ImpactEccentricty)^2-(Diameter2c/2)^2)^0.5)/(Diameter2c/2)];
 G4 = Log[(ChordatImpact/2+ImpactEccentricty+((ChordatImpact/2+ImpactEccentricty)^
          2-(Diameter2c/2)^2)^0.5)/(Diameter2c/2)];
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
BetaW = P1 * G4 / (G1 * G4 + G2 * G3);
BetaFFS = 1.08882991825178 + 0.191072321954091 * (aa / AverageAirfoilThickness) +
  -0.2275 * (aa / Diameter2c) + ((aa / AverageAirfoilThickness) - 0.2375) *
   (((aa / AverageAirfoilThickness) - 0.2375) * -0.336251716696097) +
  ((aa / AverageAirfoilThickness) - 0.2375) * (((aa / Diameter2c) - 0.256666666666667) *
     -0.642299410757611) + ((aa / Diameter2c) - 0.256666666666667) *
   (((aa / Diameter2c) - 0.256666666666667) * 0.443509615384615);
BetaCompounded = BetaBFS * BetaFFS * BetaW * BetaCFC;
BetaCompoundedAppend = AppendTo[BetaCompoundedAppend, BetaCompounded];
DeltaK = BetaCompounded * DeltaSigma * (P1 * aa)^0.5;

σResidual = 0;
If[ImpactSide == "Pressure Side", If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 1,
   σResidual = -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) +
         117 116.263407782 * aa + (aa - 0.000335) * (aa - 0.000335) * -395 474 551.758495 +
         (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * 600 608 204 212.839 + (aa - 0.000335) *
          (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -313 392 042 189 632],
        Abs[((-91.388930233299) + 163 881.618156941 * aa + (aa - 0.000335) * (aa - 0.000335) *
            -621 710 641.059832 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
            784 096 231 132.582 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
            (aa - 0.000335) * -181 928 735 061 177) - ((-51.0322827764947) + 73 621.0917592117 *
            aa + (aa - 0.000335) * (aa - 0.000335) * -186 084 270.781473 + (aa - 0.000335) *
            (aa - 0.000335) * (aa - 0.000335) * 369 136 849 298.813 + (aa - 0.000335) *
            (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -331 072 585 461 178)]/6],
      RandomReal[]] * If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1,
      If[LSPTreatment == 3, 1.25, If[LSPTreatment == 0, 0]]]]],
 If[ImpactSide == "Suction Side", If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 0.80,
    σResidual = -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) + 117 116.263407782 *
          aa + (aa - 0.000335) * (aa - 0.000335) * -395 474 551.758495 + (aa - 0.000335) *
          (aa - 0.000335) * (aa - 0.000335) * 600 608 204 212.839 + (aa - 0.000335) *
          (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -313 392 042 189 632],
        Abs[((-91.388930233299) + 163 881.618156941 * aa + (aa - 0.000335) * (aa - 0.000335) *
            -621 710 641.059832 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
            784 096 231 132.582 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
            (aa - 0.000335) * -181 928 735 061 177) - ((-51.0322827764947) + 73 621.0917592117 *
            aa + (aa - 0.000335) * (aa - 0.000335) * -186 084 270.781473 + (aa - 0.000335) *
            (aa - 0.000335) * (aa - 0.000335) * 369 136 849 298.813 + (aa - 0.000335) *
            (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -331 072 585 461 178)]/6],
      RandomReal[]] * If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1, If[
       LSPTreatment == 3, 1.25, If[LSPTreatment == 0, 0]]]]]];

Kmax = BetaCompounded * If[aa ≤ 0.00091, AxialσMax + σResidual * 10^6, AxialσMax] * (P1 * aa)^0.5;
dadN = (0.0000000000052) * ((DeltaK / 10^6)^2.5) * ((Kmax / 10^6)^0.67);
DeltaNcycles = Nincrement;
daa = dadN * DeltaNcycles;
aa = aa + daa;
If[aa === ComplexInfinity || Head[aa] === Complex, aa = Re[aa]];
If[aa ≤ DepthofPenetration,
 Diameter2c = 2 * If[aa ≤ Diameter2c/2, ((Diameter2c/2)^2 - (Diameter2c/2 - aa)^2)^0.5,
    ((Diameter2c/2)^2 - (Diameter2c/2 - Diameter2c/2)^2)^0.5], Diameter2c = 2 * aa];
Ncycles = 11 * Nincrement;
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
16 |
        nrepairs = 0;
        nreplacements = 0;
        LevelIIIDamageRandomSeed = RandomReal[];
        If[EngineIntervalInspections == 1,
         If[Ncycles == NcyclesIntervalInspection && aaD ≤ aa < aac, Nrepairs = Nrepairs + 1;
          aa = EIFSa, If[LevelIIIDamageRandomSeed ≥ 4 * 10^-5,
           If[Ncycles == NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
            aa = EIFSa], If[LevelIIIDamageRandomSeed < 4 * 10^-5 && Ncycles ==
              NcyclesIntervalInspection && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
            aa = EIFSa;
            Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
         If[LevelIIIDamageRandomSeed > 4 * 10^-5,
          If[Ncycles < NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
           aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles <
              NcyclesIntervalInspection && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
            aa = EIFSa;
            Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
         If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[NcyclesIntervalInspection * 1.5 > Ncycles >
             NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
           aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5,
           If[NcyclesIntervalInspection * 1.5 > Ncycles > NcyclesIntervalInspection && aa ≥ aac,
            NEngineReplacements = NEngineReplacements + 1;
            aa = EIFSa;
            Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
         If[Ncycles == NcyclesIntervalInspection * 1.5 && aaD ≤ aa < aac, Nrepairs = Nrepairs + 1;
          aa = EIFSa, If[LevelIIIDamageRandomSeed ≥ 4 * 10^-5,
           If[Ncycles == NcyclesIntervalInspection * 1.5 && aa ≥ aac, Nreplacements = Nreplacements + 1;
            aa = EIFSa], If[LevelIIIDamageRandomSeed < 4 * 10^-5 && Ncycles ==
              NcyclesIntervalInspection * 1.5 && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
            aa = EIFSa;
            Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
         If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[Ntakeoffs * 2 * 60 * 60 * StressFrequency > Ncycles >
             NcyclesIntervalInspection * 1.5 && aa ≥ aac, Nreplacements = Nreplacements + 1;
           aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles >
              NcyclesIntervalInspection * 1.5 && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
            aa = EIFSa;
            Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]],
         If[EngineIntervalInspections == 0, If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[Ncycles <
              Ntakeoffs * 2 * 60 * 60 * StressFrequency && aa ≥ aac, Nreplacements = Nreplacements + 1;
            aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles < Ntakeoffs * 2 *
                60 * 60 * StressFrequency && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
             aa = EIFSa;
             Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]]];
        If[aac ≤ aa || Ntakeoffs * 2 * 60 * 60 * StressFrequency == Ncycles,
         Print["ith takeoff during Natural Growth when a >= aac or
            Ncycles = Ntakeoffs*2*60*60*StressFrequency: ", i1];
         i = i + 1;
         NcyclesatbladeReplacement = AppendTo[NcyclesatbladeReplacement, Ncycles];
         NcyclesatbladeReplacementTotal = AppendTo[NcyclesatbladeReplacementTotal, Ncycles];
         NbladeReplacements = NbladeReplacements + 1;
         Ncycles = 0,
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
  aaAppended = AppendTo[aaAppended, aa];
  NcylesAppended = AppendTo[NcylesAppended, Ncycles];
  NcyclesAppendedTotal =
   AppendTo[NcyclesAppendedTotal, Transpose[{NcylesAppended, aaAppended}]];
 ],
 {ii, Impacti + 1, Ntakeoffs}];

If[Nrepairs > 0, CostIncreaseDuetoEngineEfficiencyDregradation =
  (0.005 * RandomReal[] * 0.005) * 2 * CostofFuelperFlight * (Ntakeoffs - Impacti)];
Print["CostIncreaseDuetoEngineEfficiencyDregradation: ",
 CostIncreaseDuetoEngineEfficiencyDregradation];


Print["Nrepairs: ", Nrepairs];
Print["Nreplacements: ", Nreplacements];
Print["NEnginereplacements: ", NEnginereplacements],

If[(XpercentChord > 0.80 || XpercentChord < 0.05) && ChordatImpact > 0,
 Print["Check that Edge Crack Beta Solution is triggered"];

 aac50thpercentilethroughflaw = 0.018372160295978216;
 aac = aac50thpercentilethroughflaw;

 (*aac=(aac/.Extract[FindRoot[
       {(1.12-0.231*(1000000*aac/ChordatImpact)+10.550*(1000000*aac/ChordatImpact)^2-
            21.710*(1000000*aac/ChordatImpact)^3+30.382*(1000000*aac/ChordatImpact)^4)*
          (1000000*Pi*aac)^0.5-(KMaxatFracture)/((Axialσ/10^6))},{aac,0.1}],1])*1000000;

 aacAppended=AppendTo[aacAppended,aac];
 Print["aacAppended: ",aacAppended];*)

 (*Nf=Quiet[NIntegrate[
     1/(0.0000000000052)*1/((((1.12-0.231*(aa/ChordatImpact)+10.550*(aa/ChordatImpact)^2-
               21.710*(aa/ChordatImpact)^3+30.382*(aa/ChordatImpact)^4)*(DeltaSigma/10^6)*
            (Pi*aa)^0.5)^2.5)*(KMaxatFracture)^0.67),{aa,aa0,aac}]];*)

 Print["aac: ", aac];

 Do[
  BetaCompounded =
   1.12 - 0.231 * (aaNaturalGrowth / ChordatImpact) + 10.550 * (aaNaturalGrowth / ChordatImpact)^2 -
    21.710 * (aaNaturalGrowth / ChordatImpact)^3 + 30.382 * (aaNaturalGrowth / ChordatImpact)^4;
  BetaCompoundedAppendNaturalGrowth = AppendTo[BetaCompoundedAppendNaturalGrowth,
    BetaCompounded];
  DeltaK = BetaCompounded * DeltaSigma * (Pi * aaNaturalGrowth)^0.5;

  σResidual = 0;
  If[ImpactSide == "Pressure Side", If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 1,
     σResidual = -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) +
           117116.263407782 * aaNaturalGrowth + (aaNaturalGrowth - 0.000335) *
            (aaNaturalGrowth - 0.000335) * -395474551.758495 + (aaNaturalGrowth - 0.000335) *
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
18 |
                (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * 600 608 204 212.839 +
              (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
               (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * - 313 392 042 189 632],
          Abs[((-91.388930233299) + 163 881.618156941 * aaNaturalGrowth +
                (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * - 621 710 641.059832 +
                (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
                 (aaNaturalGrowth - 0.000335) * 784 096 231 132.582 +
                (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth -
                   0.000335) * (aaNaturalGrowth - 0.000335) * - 181 928 735 061 177) -
              ((-51.0322827764947) + 73 621.0917592117 * aaNaturalGrowth +
                (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * - 186 084 270.781473 +
                (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
                 (aaNaturalGrowth - 0.000335) * 369 136 849 298.813 + (aaNaturalGrowth -
                   0.000335) * (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
                 (aaNaturalGrowth - 0.000335) * - 331 072 585 461 178)] / 6], RandomReal[]] *
       If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1, If[LSPTreatment == 3,
          1.25, If[LSPTreatment == 0, 0]]]]],
    If[ImpactSide == "Suction Side", If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 0.80,
       σResidual = -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) +
             117 116.263407782 * aaNaturalGrowth + (aaNaturalGrowth - 0.000335) *
              (aaNaturalGrowth - 0.000335) * - 395 474 551.758495 + (aaNaturalGrowth - 0.000335) *
              (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * 600 608 204 212.839 +
             (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
              (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * - 313 392 042 189 632],
           Abs[((-91.388930233299) + 163 881.618156941 * aaNaturalGrowth +
                (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
                 - 621 710 641.059832 + (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth -
                   0.000335) * (aaNaturalGrowth - 0.000335) * 784 096 231 132.582 +
                (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
                 (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
                 - 181 928 735 061 177) - ((-51.0322827764947) + 73 621.0917592117 *
                aaNaturalGrowth + (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
                 - 186 084 270.781473 + (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth -
                   0.000335) * (aaNaturalGrowth - 0.000335) * 369 136 849 298.813 +
                (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth -
                   0.000335) * (aaNaturalGrowth - 0.000335) * - 331 072 585 461 178)] / 6],
           RandomReal[]] * If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1,
           If[LSPTreatment == 3, 1.25, If[LSPTreatment == 0, 0]]]]]];

Kmax = BetaCompounded * If[aaNaturalGrowth ≤ 0.00091,
    AxialσMax + σResidual * 10^6, AxialσMax] * (PI * aaNaturalGrowth)^0.5;
dadN = (0.0000000000052) * ((DeltaK / 10^6)^2.5) * ((Kmax / 10^6)^0.67);
DeltaNcycles = Nincrement;
daa = dadN * DeltaNcycles;
aaNaturalGrowth = aaNaturalGrowth + daa;
If[aaNaturalGrowth === ComplexInfinity || Head[aaNaturalGrowth] === Complex,
 aaNaturalGrowth = Re[aaNaturalGrowth]];
Diameter2cNaturalGrowth = 2 * aaNaturalGrowth;
NcyclesNaturalGrowth = 11 * Nincrement;
aaAppendedNaturalGrowth = AppendTo[aaAppendedNaturalGrowth, aaNaturalGrowth];
NcyclesNaturalGrowthAppended =

Printed by Wolfram Mathematica Student Edition
```

Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
  AppendTo[NcyclesNaturalGrowthAppended, NcyclesNaturalGrowth];
 If[aaNaturalGrowth ≥ aac || Ntakeoffs * 2 * 60 * 60 * StressFrequency == Ncycles,
  Print["ith takeoff during Natural Growth when a >= aac or
     Ncycles = Ntakeoffs*2*60*60*StressFrequency: ", ii];
  Goto[Next4]],
 {ii, 1, Ntakeoffs}];
Label[Next4];


Do[
 BetaCompounded = 1.12 - 0.231 * (aa / ChordatImpact) + 10.550 * (aa / ChordatImpact)^2 -
   21.710 * (aa / ChordatImpact)^3 + 30.382 * (aa / ChordatImpact)^4;
 BetaCompoundedAppend = AppendTo[BetaCompoundedAppend, BetaCompounded];
 DeltaK = BetaCompounded * DeltaSigma * (Pi * aa)^0.5;

 σResidual = 0;
 If[ImpactSide == "Pressure Side",
  If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 1, σResidual =
     -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) + 117 116.263407782 * aa + (aa -
               0.000335) * (aa - 0.000335) * -395 474 551.758495 + (aa - 0.000335) * (aa - 0.000335) *
             (aa - 0.000335) * 600 608 204 212.839 + (aa - 0.000335) * (aa - 0.000335) *
             (aa - 0.000335) * (aa - 0.000335) * -313 392 042 189 632],
          Abs[((-91.388930233299) + 163 881.618156941 * aa + (aa - 0.000335) * (aa - 0.000335) *
               -621 710 641.059832 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
               784 096 231 132.582 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
               (aa - 0.000335) * -181 928 735 061 177) - ((-51.0322827764947) + 73 621.0917592117 *
               aa + (aa - 0.000335) * (aa - 0.000335) * -186 084 270.781473 + (aa - 0.000335) *
               (aa - 0.000335) * (aa - 0.000335) * 369 136 849 298.813 + (aa - 0.000335) *
               (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -331 072 585 461 178)] / 6],
        RandomReal[]] * If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1, If[
         LSPTreatment == 3, 1.25, If[LSPTreatment == 0, 0]]]]],
  If[ImpactSide == "Suction Side", If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 0.80,
     σResidual = -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) + 117 116.263407782 *
             aa + (aa - 0.000335) * (aa - 0.000335) * -395 474 551.758495 + (aa - 0.000335) *
             (aa - 0.000335) * (aa - 0.000335) * 600 608 204 212.839 + (aa - 0.000335) *
             (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -313 392 042 189 632],
          Abs[((-91.388930233299) + 163 881.618156941 * aa + (aa - 0.000335) * (aa - 0.000335) *
               -621 710 641.059832 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
               784 096 231 132.582 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * (aa -
                 0.000335) * -181 928 735 061 177) - ((-51.0322827764947) + 73 621.0917592117 *
               aa + (aa - 0.000335) * (aa - 0.000335) * -186 084 270.781473 + (aa - 0.000335) *
               (aa - 0.000335) * (aa - 0.000335) * 369 136 849 298.813 + (aa - 0.000335) *
               (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -331 072 585 461 178)] / 6],
        RandomReal[]] * If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1,
        If[LSPTreatment == 3, 1.25, If[LSPTreatment == 0, 0]]]]]];

 Kmax = BetaCompounded * If[aa ≤ 0.00091, AxialσMax + σResidual * 10^6, AxialσMax] * (Pi * aa)^0.5;
 dadN = (0.0000000000052) * ((DeltaK / 10^6)^2.5) * ((Kmax / 10^6)^0.67);
 DeltaNcycles = Nincrement;
 daa = dadN * DeltaNcycles;
 aa = aa + daa;
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
20 |
    If[aa === ComplexInfinity || Head[aa] === Complex, aa = Re[aa]];
    Diameter2c = 2 * aa;
    Ncycles = 11 * Nincrement;
    aaAppended = AppendTo[aaAppended, aa];
    NcylesAppended = AppendTo[NcylesAppended, Ncycles];
    LevelIIIDamageRandomSeed = RandomReal[];
    If[EngineIntervalInspections == 1,
     If[Ncycles == NcyclesIntervalInspection && aaD ≤ aa < aac, Nrepairs = Nrepairs + 1;
      aa = EIFSa, If[LevelIIIDamageRandomSeed ≥ 4 * 10^-5,
       If[Ncycles == NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
        aa = EIFSa], If[LevelIIIDamageRandomSeed < 4 * 10^-5 && Ncycles ==
          NcyclesIntervalInspection && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
        aa = EIFSa;
        Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
     If[LevelIIIDamageRandomSeed > 4 * 10^-5,
      If[Ncycles < NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
       aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles <
          NcyclesIntervalInspection && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
        aa = EIFSa;
        Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
     If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[NcyclesIntervalInspection * 1.5 > Ncycles >
         NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
       aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5,
       If[NcyclesIntervalInspection * 1.5 > Ncycles > NcyclesIntervalInspection && aa ≥ aac,
        NEngineReplacements = NEngineReplacements + 1;
        aa = EIFSa;
        Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
     If[Ncycles == NcyclesIntervalInspection * 1.5 && aaD ≤ aa < aac, Nrepairs = Nrepairs + 1;
      aa = EIFSa, If[LevelIIIDamageRandomSeed ≥ 4 * 10^-5,
       If[Ncycles == NcyclesIntervalInspection * 1.5 && aa ≥ aac, Nreplacements = Nreplacements + 1;
        aa = EIFSa], If[LevelIIIDamageRandomSeed < 4 * 10^-5 &&
         Ncycles == NcyclesIntervalInspection * 1.5 && aa ≥ aac,
        NEngineReplacements = NEngineReplacements + 1;
        aa = EIFSa;
        Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
     If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[Ntakeoffs * 2 * 60 * StressFrequency > Ncycles >
         NcyclesIntervalInspection * 1.5 && aa ≥ aac, Nreplacements = Nreplacements + 1;
       aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5,
       If[Ncycles > NcyclesIntervalInspection * 1.5 && aa ≥ aac,
        NEngineReplacements = NEngineReplacements + 1;
        aa = EIFSa;
        Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]],
     If[EngineIntervalInspections == 0, If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[Ncycles <
         Ntakeoffs * 2 * 60 * 60 * StressFrequency && aa ≥ aac, Nreplacements = Nreplacements + 1;
       aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles < Ntakeoffs * 2 *
            60 * 60 * StressFrequency && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
        aa = EIFSa;
        Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]]],
    {11, 1, Impact1}];

   Print["BetaCompounded: ", BetaCompounded];
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
Print["aa Check: ", aa];

If[aa ≤ DepthofPenetration, aa = DepthofPenetration;
 Diameter2c = 2 * If[aa ≤ FodDiam/2, ((FodDiam/2)^2 - (FodDiam/2 - aa)^2)^0.5,
    ((FodDiam/2)^2 - (FodDiam/2 - FodDiam/2)^2)^0.5], Diameter2c = 2 * aa];

(*aaOregressed=aa;
Print["Regressed aa before Do Loop starts:",aaOregressed];

Do[
 BetaCompounded=
  1.12-0.231*(aaOregressed/ChordatImpact)+10.550*(aaOregressed/ChordatImpact)^2-
   21.710*(aaOregressed/ChordatImpact)^3+30.382*(aaOregressed/ChordatImpact)^4;
 DeltaK=BetaCompounded*DeltaSigma*(Pi*aaOregressed)^0.5;
 Kmax=BetaCompounded*AxialσMax*(Pi*aaOregressed)^0.5;
 aaOregressed=
  aaOregressed-(0.0000000000052)*((DeltaK/10^6)^2.5)*((Kmax/10^6)^0.67)*Nincrement,
 {ii,1,Impacti+1}];

Print["aaOregressed: ",aaOregressed," aa at Impact: ",aa];

If[aaOregressed===ComplexInfinity||Head[aaOregressed]===Complex,
 Goto[Next6],aaOregressedAppended=AppendTo[aaOregressedAppended,aaOregressed]];

Label[Next6];*)

Print["Nrepairs: ", Nrepairs];
Print["Nreplacements: ", Nreplacements];

aaAppended = AppendTo[aaAppended, aa];
NcyclesAppended = AppendTo[NcyclesAppended, Ncycles];
Print["Ncycles Between Initial Growth and Impact: ", Ncycles];
Print["NcyclesatImpact: ", Impacti * Nincrement];

Do[

 BetaCompounded = 1.12 - 0.231 * (aa / ChordatImpact) + 10.550 * (aa / ChordatImpact)^2 -
   21.710 * (aa / ChordatImpact)^3 + 30.382 * (aa / ChordatImpact)^4;
 BetaCompoundedAppend = AppendTo[BetaCompoundedAppend, BetaCompounded];
 DeltaK = BetaCompounded * DeltaSigma * (Pi * aa)^0.5;

 σResidual = 0;
 If[ImpactSide == "Pressure Side",
  If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 1, σResidual =
     -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) + 117 116.263407782 * aa + (aa -
               0.000335) * (aa - 0.000335) * -395 474 551.758495 + (aa - 0.000335) * (aa - 0.000335) *
            (aa - 0.000335) * 600 608 204 212.839 + (aa - 0.000335) * (aa - 0.000335) *
            (aa - 0.000335) * (aa - 0.000335) * -313 392 042 189 632],
         Abs[((-91.388930233299) + 163 881.618156941 * aa + (aa - 0.000335) * (aa - 0.000335) *
             -621 710 641.059832 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
             784 096 231 132.582 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
22 |
                    (aa - 0.000335) * -181 928 735 061 177) - ((-51.0322827764947) + 73 621.0917592117 *
                   aa + (aa - 0.000335) * (aa - 0.000335) * -186 084 270.781473 + (aa - 0.000335) *
                   (aa - 0.000335) * (aa - 0.000335) * 369 136 849 298.813 + (aa - 0.000335) *
                   (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -331 072 585 461 178)] / 6],
          RandomReal[]] * If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1, If[
           LSPTreatment == 3, 1.25, If[LSPTreatment == 0, 0]]]]],
       If[ImpactSide == "Suction Side", If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 0.80,
          σResidual = -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) + 117 116.263407782 *
                 aa + (aa - 0.000335) * (aa - 0.000335) * -395 474 551.758495 + (aa - 0.000335) *
                 (aa - 0.000335) * (aa - 0.000335) * 600 608 204 212.839 + (aa - 0.000335) *
                 (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -313 392 042 189 632],
           Abs[((-91.388930233299) + 163 881.618156941 * aa + (aa - 0.000335) * (aa - 0.000335) *
                 -621 710 641.059832 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
                 784 096 231 132.582 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * (aa -
                   0.000335) * -181 928 735 061 177) - ((-51.0322827764947) + 73 621.0917592117 *
                 aa + (aa - 0.000335) * (aa - 0.000335) * -186 084 270.781473 + (aa - 0.000335) *
                 (aa - 0.000335) * (aa - 0.000335) * 369 136 849 298.813 + (aa - 0.000335) *
                 (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -331 072 585 461 178)] / 6],
          RandomReal[]] * If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1,
          If[LSPTreatment == 3, 1.25, If[LSPTreatment == 0, 0]]]]]];

Kmax = BetaCompounded * If[aa ≤ 0.00091, AxialσMax + σResidual * 10^6, AxialσMax] * (Pi * aa)^0.5;
dadN = (0.0000000000052) * ((DeltaK / 10^6)^2.5) * ((Kmax / 10^6)^0.67);
DeltaNcycles = Nincrement;
daa = dadN * DeltaNcycles;
aa = aa + daa;
If[aa === ComplexInfinity || Head[aa] === Complex, aa = Re[aa]];
Ncycles = 11 * Nincrement;
LevelIIIDamageRandomSeed = RandomReal[];
If[EngineIntervalInspections == 1,
 If[Ncycles == NcyclesIntervalInspection && aaD ≤ aa < aac, Nrepairs = Nrepairs + 1;
  aa = EIFSa, If[LevelIIIDamageRandomSeed ≥ 4 * 10^-5,
   If[Ncycles == NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
    aa = EIFSa], If[LevelIIIDamageRandomSeed < 4 * 10^-5 && Ncycles ==
      NcyclesIntervalInspection && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
 If[LevelIIIDamageRandomSeed > 4 * 10^-5,
  If[Ncycles < NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
   aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles <
     NcyclesIntervalInspection && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
 If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[NcyclesIntervalInspection * 1.5 > Ncycles >
    NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
   aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5,
   If[NcyclesIntervalInspection * 1.5 > Ncycles > NcyclesIntervalInspection && aa ≥ aac,
    NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
 If[Ncycles == NcyclesIntervalInspection * 1.5 && aaD ≤ aa < aac, Nrepairs = Nrepairs + 1;

Printed by Wolfram Mathematica Student Edition
```

Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
      aa = EIFSa, If[LevelIIIDamageRandomSeed ≥ 4 * 10^-5,
       If[Ncycles = NcyclesIntervalInspection * 1.5 && aa ≥ aac, Nreplacements = Nreplacements + 1;
        aa = EIFSa], If[LevelIIIDamageRandomSeed < 4 * 10^-5 &&
         Ncycles = NcyclesIntervalInspection * 1.5 && aa ≥ aac,
        NEnginereplacements = NEnginereplacements + 1;
        aa = EIFSa;
        Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]]];
      If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[Ntakeoffs * 2 * 60 * 60 * StressFrequency > Ncycles >
          NcyclesIntervalInspection * 1.5 && aa ≥ aac, Nreplacements = Nreplacements + 1;
        aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5,
        If[Ncycles > NcyclesIntervalInspection * 1.5 && aa ≥ aac,
         NEnginereplacements = NEnginereplacements + 1;
         aa = EIFSa;
         Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]],
       If[EngineIntervalInspections = 0, If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[Ncycles <
            Ntakeoffs * 2 * 60 * 60 * StressFrequency && aa ≥ aac, Nreplacements = Nreplacements + 1;
         aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles < Ntakeoffs * 2 *
              60 * 60 * StressFrequency && aa ≥ aac, NEnginereplacements = NEnginereplacements + 1;
          aa = EIFSa;
          Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]]]];
     If[aac ≤ aa || Ntakeoffs * 2 * 60 * 60 * StressFrequency = Ncycles,
      Print["ith takeoff during Natural Growth when a >= aac or
         Ncycles = Ntakeoffs*2*60*60*StressFrequency: ", ii];
      i = i + 1;
      Ncyclesatbladereplacement = AppendTo[Ncyclesatbladereplacement, Ncycles];
      NcyclesatbladereplacementTotal = AppendTo[NcyclesatbladereplacementTotal, Ncycles];
      Nbladereplacements = Nbladereplacements + 1;
      Ncycles = 0,
      aaAppended = AppendTo[aaAppended, aa];
      NcylesAppended = AppendTo[NcylesAppended, Ncycles]],
     {ii, Impacti + 1, Ntakeoffs}];

    If[Nrepairs > 0, CostIncreaseDuetoEngineEfficiencyDregradation =
      (0.005 + RandomReal[] * 0.005) * 2 * CostofFuelperFlight * (Ntakeoffs - Impacti)];
    Print["CostIncreaseDuetoEngineEfficiencyDregradation: ",
     CostIncreaseDuetoEngineEfficiencyDregradation];

    Print["Nrepairs: ", Nrepairs];
    Print["Nreplacements: ", Nreplacements];
    Print["NEnginereplacements: ", NEnginereplacements];


    Goto[End1]]], Goto[Next5]], Goto[Next5]]; Label[Next5], {i, 1, Ntakeoffs}];

Label[End1];

LifeTimerepairs = Nrepairs - LifeTimerepairs;
Print["LifeTimerepairs: ", LifeTimerepairs];
LifeTimereplacements = Nreplacements - LifeTimereplacements;
Print["LifeTimereplacements: ", LifeTimereplacements];
LifeTimeEnginereplacements = NEnginereplacements - LifeTimeEnginereplacements;
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
24 |
    Print["LifeTimeEngineReplacements: ", LifeTimeEngineReplacements];

    LifeTimeFuelCostDueToEfficiencyDegradation = CostIncreaseDuetoEngineEfficiencyDregradation;


    NRepairsAppended = AppendTo[NRepairsAppended, LifeTimeRepairs];
    NReplacementsAppended = AppendTo[NReplacementsAppended, LifeTimeReplacements];
    NEngineReplacementsAppended =
     AppendTo[NEngineReplacementsAppended, LifeTimeEngineReplacements];

    (* Life Time Scheduled Corrective Action Labor Costing Models *)

    SCHCALRepairs = LifeTimeRepairs * (FLTS * NA * QTY * AVEFLTHR * RECMHRSRepairs * BurdenedDIRLAB) /
        (NcyclesIntervalInspection / (60 * 60 * StressFrequency));
    SCHCALRepairsAppended = AppendTo[SCHCALRepairsAppended, SCHCALRepairs];

    SCHEDCAMATRepairs = LifeTimeRepairs * (FLTS * NA * QTY * AVEFLTHR * SCHCAMATRepairs) /
        (NcyclesIntervalInspection / (60 * 60 * StressFrequency));
    Print["SCHEDCAMATRepairs: ", SCHEDCAMATRepairs];
    SCHEDCAMATRepairsAppended = AppendTo[SCHEDCAMATRepairsAppended, SCHEDCAMATRepairs];

    LifeTimeRepairsCosts = SCHCALRepairs + SCHEDCAMATRepairs;
    LifeTimeRepairsCostsAppended = AppendTo[LifeTimeRepairsCostsAppended, LifeTimeRepairsCosts];

    (* Component Replacements and Associated Labor *)
    LRReplacements =
     LifeTimeReplacements * (FLTS * NA * QTY * AVEFLTHR * RECMHRSReplacements * BurdenedDIRLAB) /
        (NcyclesIntervalInspection / (60 * 60 * StressFrequency));
    LRReplacementsAppended = AppendTo[LRReplacementsAppended, LRReplacements];

    MATReplacements = LifeTimeReplacements * (FLTS * NA * QTY * AVEFLTHR * CAMATReplacements) /
        (NcyclesIntervalInspection / (60 * 60 * StressFrequency));
    CAMATReplacementsAppended = AppendTo[CAMATReplacementsAppended, MATReplacements];

    LifeTimeReplacementsCosts = LRReplacements + MATReplacements;
    LifeTimeReplacementsCostsAppended =
     AppendTo[LifeTimeReplacementsCostsAppended, LifeTimeReplacementsCosts];

    (* Engine Replacements *)
    LEngineReplacements =
     LifeTimeEngineReplacements * (FLTS * NA * QTY * AVEFLTHR * MHRSEngineReplacements * BurdenedDIRLAB) /
        (NcyclesIntervalInspection / (60 * 60 * StressFrequency));
    LEngineReplacementsAppended = AppendTo[LEngineReplacementsAppended, LEngineReplacements];

    MATEngineReplacements =
     LifeTimeEngineReplacements * (FLTS * NA * QTY * AVEFLTHR * CAMATEngineReplacements) /
        (NcyclesIntervalInspection / (60 * 60 * StressFrequency));
    MATEngineReplacementsAppended = AppendTo[MATEngineReplacementsAppended, MATEngineReplacements];

    TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisitionAppended =
     AppendTo[TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisitionAppended,
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
  TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisition];

Print["LifeTimeFuelCostDueToEfficiencyDegradation: ",
 LifeTimeFuelCostDueToEfficiencyDegradation];

Print["LifeTimeCostofScheduledInspections: ", SCHINCOST];
Print["LifeTimeCostofScheduledRepairLabor: ", SCHCALrepairs];
Print["LifeTimeCostofUnscheduledReplacementLabor: ", Lreplacements];
Print["LifeTimeCostofScheduledRepairMaterial: ", SCHEDCAMATrepairs];
Print["LifeTimeCostofUnscheduledRepairMaterial: ", MATreplacements];
Print["TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisition: ",
 TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisition];
Print["LifeTimeCostofEngineReplacementLabor: ", LEnginereplacements];
Print["LifeTimeCostofEngineReplacement: ", MATEnginereplacements];
Print["LifeTimeFuelCostDueToEfficiencyDegradation: ",
 LifeTimeFuelCostDueToEfficiencyDegradation];

TotalLifeTimeCosts = SCHINCOST + SCHCALrepairs + Lreplacements + SCHEDCAMATrepairs +
  MATreplacements + TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisition +
  LEnginereplacements + MATEnginereplacements + NA * LifeTimeFuelCostDueToEfficiencyDegradation;
Print["TotalLifeTimeCosts: ", TotalLifeTimeCosts];
TotalLifeTimeCostsAppended = AppendTo[TotalLifeTimeCostsAppended, TotalLifeTimeCosts];



NbladereplacementsAppended = AppendTo[NbladereplacementsAppended, Nbladereplacements];

aavsNcycles = Transpose[{NcylesAppended, aaAppended}];
Print[
 "DOE Settings for Scan Rate, Runway Sweeps, Scan Probability of Detection, LSP Condition and
   Determination if Engine Interval Inspections Take Place: ", CrackGrowthModel[[i1]]];
Print[ListPlot[aavsNcycles, GridLines → Automatic, PlotLegends →
   "Crack Growth a Including Post Impact",
  AxesLabel → {"Stress Cycles (N)", "CrackDepth a (meters)"}]];
aavsNcyclesNaturalGrowth = Transpose[{NcyclesNaturalGrowthAppended, aaAppendedNaturalGrowth}];
Print[ListPlot[aavsNcyclesNaturalGrowth, GridLines → Automatic, PlotLegends →
   "Natural Crack Growth a", AxesLabel → {"Stress Cycles (N)", "CrackDepth a (meters)"}]];
Print[ListPlot[{aavsNcycles, aavsNcyclesNaturalGrowth}, GridLines → Automatic,
  PlotLegends → {"Crack Growth a Including Post Impact", "Natural Crack Growth a"},
  AxesLabel → {"Stress Cycles (N)", "CrackDepth a (meters)"}]];

(*Print["Nbladereplacements: ",Nbladereplacements];
Print["Ncyclesatbladereplacement: ",Ncyclesatbladereplacement];*)

(*Print["aavsNcycles: ",aavsNcycles];
Print["NcyclesAppendedTotal: ",NcyclesAppendedTotal];*)

ClearAll[Ncycles, NcyclesNaturalGrowth];
Label[End2],
{n, 1, 101}];
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
26 |
    (*Print["aa0regressedAppended: ",aa0regressedAppended]
      aa0regressed90thPercentile=Quantile[aa0regressedAppended,0.90];
    Print["aa0regressed90thPercentile: ",aa0regressed90thPercentile]
      Print["NbladereplacementsAppended: ",NbladereplacementsAppended]
      Print["NcyclesatbladereplacemenT: ",NcyclesatbladereplacemenT]
      Histogram[NbladereplacementsAppended,AxesLabel→{"NbladereplacementsAppended","Frequency"}]
      Histogram[NcyclesatbladereplacemenT,AxesLabel→{"NcyclesatbladereplacemenT","Frequency"}]
      Ncycles90thPercentileInspection=Quantile[NcyclesatbladereplacemenT,0.90]/2;
    Print["Ncycles90thPercentileInspection: ",Ncycles90thPercentileInspection]*)
    Print[
     "DOE Settings for Scan Rate, Runway Swoops, Scan Probability of Detection, LSP Condition and
       Determination if Engine Interval Inspections Take Place: ", CrackGrowthModel[[i1]]];
    Print["Nimpacts: ", Nimpacts];
    Print["Nrepairs: ", Nrepairs];
    Print["Nreplacements: ", Nreplacements];
    Print["NEnginereplacements: ", NEnginereplacements];
    Print["NrepairsAppended: ", NrepairsAppended];
    Print["NreplacementsAppended: ", NreplacementsAppended];
    Print["NEnginereplacementsAppended: ", NEnginereplacementsAppended];
    Print["SCHCALrepairsAppended: ", SCHCALrepairsAppended];
    Print["LreplacementsAppended: ", LreplacementsAppended];
    Print["SCHEDCAMATrepairsAppended: ", SCHEDCAMATrepairsAppended];
    Print["CAMATreplacementsAppended: ", CAMATreplacementsAppended];
    Print["LEnginereplacementsAppended: ", LEnginereplacementsAppended];
    Print["MATEnginereplacementsAppended: ", MATEnginereplacementsAppended];
    Print["TotalLifeTimeCostsAppended: ", TotalLifeTimeCostsAppended];
    Histogram[TotalLifeTimeCostsAppended,
     AxesLabel → {"TotalLifeTimeCostsAppended Using Automated Inspections", "Frequency"}];
    NrepairsAppendedDOE = AppendTo[NrepairsAppendedDOE, NrepairsAppended];
    NreplacementsAppendedDOE = AppendTo[NreplacementsAppendedDOE, NreplacementsAppended];
    NEnginereplacementsAppendedDOE =
     AppendTo[NEnginereplacementsAppendedDOE, NEnginereplacementsAppended];
    NimpactsAppendedDOE = AppendTo[NimpactsAppendedDOE, Nimpacts];
    TotalLifeTimeCostsAppendedDOE =
     AppendTo[TotalLifeTimeCostsAppendedDOE, TotalLifeTimeCostsAppended];
    TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisitionAppendedDOE =
     AppendTo[TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisitionAppendedDOE,
      TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisitionAppended];
    CAMATreplacementsAppendedDOE = AppendTo[CAMATreplacementsAppendedDOE, CAMATreplacementsAppended];
    LifeTimereplacementscostsAppendedDOE =
     AppendTo[LifeTimereplacementscostsAppendedDOE, LifeTimereplacementscostsAppended];
    LifeTimerepairscostsAppendedDOE =
     AppendTo[LifeTimerepairscostsAppendedDOE, LifeTimerepairscostsAppended],
    {i1, 1, InputFileLength}]

   Print["DOE Settings for Scan Rate, Runway Swoops, Scan Probability of Detection, LSP Condition and
       Determination if Engine Interval Inspections Take Place: ", CrackGrowthModel];
   Print["TotalLifeTimeCostsAppendedDOE: ", TotalLifeTimeCostsAppendedDOE]
   Print["NrepairsAppendedDOE: ", NrepairsAppendedDOE]
   Print["NreplacementsAppendedDOE: ", NreplacementsAppendedDOE]
   Print["NEnginereplacementsAppendedDOE: ", NEnginereplacementsAppendedDOE]
```


Crack Growth and LCC Code – Automated Inspections, 1.33 mm Steel FOD (cont.)

```
Print["NimpactsAppendedDOE: ", NimpactsAppendedDOE]
Print["TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisitionAppendedDOE: ",
 TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisitionAppendedDOE]
Print["LifeTimereplacementscostsAppendedDOE: ", LifeTimereplacementscostsAppendedDOE]
Print["LifeTimerepairscostsAppendedDOE: ", LifeTimerepairscostsAppendedDOE]


Show[Histogram[Table[NrepairsAppendedDOE[[1]], {1, 1, Length[NrepairsAppendedDOE]}],
  PlotRange → All, PlotLabel → "NrepairsAppended", ChartStyle → "Pastel",
  AxesLabel → {"NrepairsAppended", "Frequency"}]]

Show[Histogram[Table[NreplacementsAppendedDOE[[1]], {1, 1, Length[NreplacementsAppendedDOE]}],
  PlotRange → All, PlotLabel → "NreplacementsAppended", ChartStyle → "Pastel",
  AxesLabel → {"NreplacementsAppended", "Frequency"}]]

Show[Histogram[
  Table[NEnginereplacementsAppendedDOE[[1]], {1, 1, Length[NEnginereplacementsAppendedDOE]}],
  PlotRange → All, PlotLabel → "NEnginereplacementsAppended", ChartStyle → "Pastel",
  AxesLabel → {"NEnginereplacementsAppended", "Frequency"}]]

Show[Histogram[Table[NimpactsAppendedDOE[[1]], {1, 1, Length[NimpactsAppendedDOE]}],
  PlotRange → All, PlotLabel → "NimpactsAppended", ChartStyle → "Pastel",
  AxesLabel → {"NimpactsAppended", "Frequency"}]]

Show[Histogram[
  Table[TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisitionAppendedDOE[[1]],
   {1, 1, Length[TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisitionAppendedDOE]}],
  PlotRange → All, PlotLabel →
   "TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisitionAppended",
  ChartStyle → "Pastel", AxesLabel →
   {"TotalCostofAutomatedInspectionsandRunwaySweepandSweeperAcquisitionAppended", "Frequency"}]]

Show[
 Histogram[Table[TotalLifeTimeCostsAppendedDOE[[1]], {1, 1, Length[TotalLifeTimeCostsAppendedDOE]}],
  PlotRange → All, PlotLabel → "TotalLifeTimeCostsforAutomatedInspections", ChartStyle → "Pastel",
  AxesLabel → {"TotalLifeTimeCosts", "Frequency"}, ChartLegends → Automatic]]

Table[Show[Histogram[TotalLifeTimeCostsAppendedDOE[[1]], PlotRange → All,
   PlotLabel → "TotalLifeTimeCostsUsingAutomatedInspections", ChartStyle → "Pastel",
   AxesLabel → {"TotalLifeTimeCosts", "Frequency"}, ChartLegends → "DOE Setting" -> i]],
 {i, 1, Length[TotalLifeTimeCostsAppendedDOE]}]

Do[TotalLifeTimeCostsAppendedDOEAutomatedMean =
   AppendTo[TotalLifeTimeCostsAppendedDOEAutomatedMean, Mean[TotalLifeTimeCostsAppendedDOE[[i]]]];
  TotalLifeTimeCostsAppendedDOEAutomatedStd = AppendTo[TotalLifeTimeCostsAppendedDOEAutomatedStd,
    StandardDeviation[TotalLifeTimeCostsAppendedDOE[[i]]]];
  TotalLifeTimeCostsAppendedDOEAutomatedMeanvsDOESetting =
   AppendTo[TotalLifeTimeCostsAppendedDOEAutomatedMeanvsDOESetting,
    {i, Mean[TotalLifeTimeCostsAppendedDOE[[i]]]}];
  TotalLifeTimeCostsAppendedDOEAutomatedStdvsDOESetting =
   AppendTo[TotalLifeTimeCostsAppendedDOEAutomatedStdvsDOESetting,
```


Crack Growth and LCC Code – Automated Inspections, 1.33mm Steel FOD (comp.)

```
28 |
      {i, StandardDeviation[TotalLifeTimeCostsAppendedDOE[[i]]]},
    {i, 1, Length[TotalLifeTimeCostsAppendedDOE]}];

Do[NreplacementsvsDOESettingAppendedAutomatedSum =
    AppendTo[NreplacementsvsDOESettingAppendedAutomatedSum,
     {ii, Sum[NreplacementsAppendedDOE[[ii]][[i]], {i, 1, Length[NreplacementsAppendedDOE[[ii]]]}]}],
   {ii, 1, Length[NreplacementsAppendedDOE]}];

Do[NrepairsvsDOESettingAppendedAutomatedSum = AppendTo[NrepairsvsDOESettingAppendedAutomatedSum,
     {ii, Sum[NrepairsAppendedDOE[[ii]][[i]], {i, 1, Length[NrepairsAppendedDOE[[ii]]]}]}],
   {ii, 1, Length[NrepairsAppendedDOE]}];

ListPlot[TotalLifeTimeCostsAppendedDOEAutomatedMeanvsDOESetting,
 AxesLabel → {"DOE Setting", "TotalLifeTimeCosts"},
 PlotLegends → "Automated Inspections 1.33 mm Impactor 100 Replicates"]
ListPlot[TotalLifeTimeCostsAppendedDOEAutomatedStdvsDOESetting,
 AxesLabel → {"DOE Setting", "StdTotalLifeTimeCosts"},
 PlotLegends → "Automated Inspections 1.33 mm Impactor 100 Replicates"]
ListPlot[NreplacementsvsDOESettingAppendedAutomatedSum,
 AxesLabel → {"DOE Setting", "Sum of Component Replacements"},
 PlotLegends → "Automated Inspections 1.33 mm Impactor 100 Replicates"]
ListPlot[NrepairsvsDOESettingAppendedAutomatedSum,
 AxesLabel → {"DOE Setting", "Sum of Component Repairs"},
 PlotLegends → "Automated Inspections 1.33 mm Impactor 100 Replicates"]

TotalLifeTimeCostsStandardDeviationofMeans =
  StandardDeviation[TotalLifeTimeCostsAppendedDOEAutomatedMean];
StdofMeansandNumberofReplicatesList = {100, TotalLifeTimeCostsStandardDeviationofMeans};

Print["TotalLifeTimeCostsStandardDeviationofMeans: ", TotalLifeTimeCostsStandardDeviationofMeans]
Print["StdofMeansandNumberofReplicatesList: ", StdofMeansandNumberofReplicatesList]

Print["aacAppended: ", aacAppended]
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD

```
ClearSystemCache
Off[Power::infy]
"Statistics`NonlinearFit`"
AppendTo[$Path, ToFileName[{$HomeDirectory, "MathematicaRuns"}]];
CrackGrowthModel = Import[
  "CrackGrowthModelNoBoundaryLayerHeadWindandUGEqual133mmSteelInput_FullFactorialVisualInspection\
    s24Runs.csv", "CSV"]
InputFileLength = Length[CrackGrowthModel];
TotalLifeTimeCostsAppendedDOE = {};
NrepairsAppendedDOE = {};
NreplacementsAppendedDOE = {};
aacAppendedThroughFlaw = {};
aacAppendedThumbnailFlaw = {};
NEnginereplacementsAppendedDOE = {};
NimpactsAppendedDOE = {};
TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisitionAppendedDOE = {};
CAMATreplacementsAppendedDOE = {};
LifeTimerepairscostsAppendedDOE = {};
LifeTimereplacementscostsAppendedDOE = {};
TotalLifeTimeCostsAppendedDOEVisualMean = {};
TotalLifeTimeCostsAppendedDOEVisualStd = {};
TotalLifeTimeCostsAppendedDOEVisualMeanvsDOESetting = {};
TotalLifeTimeCostsAppendedDOEVisualStdvsDOESetting = {};
NreplacementsvsDOESettingAppendedVisualSum = {};
NrepairsvsDOESettingAppendedVisualSum = {};


Do[

 ClearSystemCache;
 Off[Power::infy];
 "Statistics`NonlinearFit`";
 ClearAll[Ncycles, aa, Nf];

 NbladereplacementsAppended = {};
 NcyclesatbladereplacementTotal = {};
 aa0regrossedAppended = {};
 NcyclesatbladereplacementTotal = {};
 Nimpacts = 0;
 Nrepairs = 0;
 Nreplacements = 0;
 NrepairsAppended = {};
 NreplacementsAppended = {};
 SCHCALrepairsAppended = {};
 LreplacementsAppended = {};
 SCHEDCAMATrepairsAppended = {};
 CAMATreplacementsAppended = {};
 TotalLifeTimeCostsAppended = {};
 NEnginereplacements = 0;
 NEnginereplacementsAppended = {};
 LEnginereplacementsAppended = {};
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
2 |
MATEngineReplacementsAppended = {};
NImpactsAppended = {};
LifeTimeFuelCostDueToEfficiencyDegradation = 0;
TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisitionAppended = {};
LifeTimeRepairsCostsAppended = {};
LifeTimeReplacementsCostsAppended = {};


LifeTimeRepairs = 0;
LifeTimeReplacements = 0;
SCHCALrepairs = 0;
LreplacementS = 0;
LEngineReplacements = 0;
LifeTimeEngineReplacements = 0;
LifeTimeRepairsCosts = 0;
LifeTimeReplacementsCosts = 0;

(* Runway Geometry, FOD Quantity on Runway,
FOD Density on Runway, FOD Geometry, FOD Sweeping performance factors *)
Lrunway = 3048; (* Runway Length *)
Wrunway = 76.2; (* Runway Width *)
FODCountonRunway = 25 000;
(* Ammount of continuously/evenly distributed FOD on runway *)
ρFodonRunway = 1 / 9.2903;
(* Density of continuously/evenly distributed FOD on runway *)
DParticle = 0.00133;
Dstagpoint = 0.0030 * 2 * (30)^0.5;
(* Scaled diameter of vortex stagnation point on runway *)
nrotorblades = 38;
(* Number of rotor blades on the 1st stage compressor rotor *)
Ninspections = CrackGrowthModel[[11, 1]];
(* Number of runway visual inspections *)
ηvisual = 0.80; (* Current efficacy of visual inspections at detecting FOD on the runway *)
ηsweep = 0.90; (* Current efficacy of FOD removal from sweeper on runway *)
FODBossSweptAreaperhour = 300 000;
(* FOD-Boss in the Triplex Trailer Configuration can sweep 300000 m^2/hour *)
nRunwaySweeps = CrackGrowthModel[[11, 2]]; (* Number of runway sweeps per operational day *)
TvisualInspection = 0.5; (* Length of visual inspection in hours *)
InspectionVehicleSpeed = 30; (* Given in mph *)
POD = 0.50;
Naircraft = 1200; (* Number of Aircraft that may use runway *)
Wsweeper = 7.32; (* Sweeper width *)

R = 0.80; (* Load ratio σmin/σmax *)
StressFrequency = 25; (* Load frequency, spectrum *)
Nincrement = 2 * 60 * 60 * StressFrequency; (* Number of load cycles per takeoff *)
aaD = 0.000381;
(* Minimum detectable flaw size in meters with current Non Destructive Inspection methods *)
FodDiam = 0.00133;
Ntakeoffs = 300; (* Number of takeoffs per engine Life Time,
takeoffs before 1st major teardown removal/overhaul of engine *)
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
NcyclesIntervalInspection = 27 000 000;
(* 1/2 90th percentile cycles for Life Time of 300 takeoffs *)
NIntervalInspections = 2; (* Two  Engine interval inspections,
first at 1/2 90th percentile comp. life, second at 3/4 90th percentile comp. life *)
EngineIntervalInspections = CrackGrowthModel[[11, 4]];
(* Engine Non-Destructive Inspections on (1) or off (0) *)

(* Life Cycle Costing, Inspection Labor Costs *)

FLTS = 135; (* Flights per year per Aircraft *)
NA = 1200; (* Number of Aircraft, set to 1200 for fleet total *)
QTY = 26; (* Component quantity per Aircraft, 2 sets of 13 rotor disks *)
AVEFLTHR = 2; (* Average flight hour per flight *)
BurdenedDIRLAB = 111 426.95 / 2080;
(* Burdened labor cost per hour, in 2015 US $ *)
SCHMNTINSMM = 8;
(* Scheduled Inspection Labor for 1/2 90th percentile comp. life
 and at at 3/4 90th percentile comp. life, for 2 personnel at 4 hours each *)
RECMHRSrepairs = 16;
(* Rectification Man Hours for 1/2 90th percentile
 comp. life  and at at 3/4 90th percentile comp. life, repairs,
for four personnel at 4 hours each, partial engine removal, access pannels open *)
RECMHRSreplacements = 8;
(* Rectification Man Hours for 1/2 90th percentile
 comp. life  and at at 3/4 90th percentile comp. life, replacements,
for four personnel at 2 hours each, engine is not removed, only access pannels *)
SCHCAMATrepairs = 1000; (* Scheduled Corrective Action Material *)
CAMATreplacements = 5000;
SCHINCOST = EngineIntervalInspections *
  NIntervalInspections * (FLTS * NA * QTY * AVEFLTHR * SCHMNTINSMM * BurdenedDIRLAB) /
    (NcyclesIntervalInspection / (60 * 60 * StressFrequency));
(* Burdened Labor Cost per Scheduled Inspection of Engine *)
Print["Burdened Labor Cost per Scheduled Inspection of Engine: ", SCHINCOST];
MHRSEnginereplacements = 3; (* Man Hours for an Engine Replacement *)
CAMATEnginereplacements = 2 460 000; (* Cost of Engine, 2015 US$ *)



MilitaryThrust = 79.1; (* kN *)
SpecificFuelConsumption = 77.5; (* kg/(kN·h) at MilitaryThrust *)
CurrentPriceofJetFuel = 1.02; (* Price per gallon, IATA current price *)
CostofFuelperFlight = MilitaryThrust * SpecificFuelConsumption * 2 * 0.3261 * CurrentPriceofJetFuel;
(* two hour flight,  0.3261 is conversion from kg to gallons of fuel *)


LSPTreatment = CrackGrowthModel[[11, 3]];
(* 3 if LSP Treatment is considered or 0 if not *)

CostofAcquisitionLSPMachine = 66 976 + RandomReal[] * (109 702 - 66 976);
(* LSP cost data from Shukla's paper *)
LSPProcessTime = 1 + RandomReal[] * 9;
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
(* Process time in minutes, LSP data from Shukla's paper *)
LSPCleanUpTime = 5; (* Cleanup time in minutes, LSP data from Shukla's paper *)
LSPChageOverTime = 10;
(* Change over time between LSP processes in minutes, LSP data from Shukla's paper *)
LSPProcessCost = QTY * NA * (nrotorblades *
    (LSPTreatment * LSPProcessTime + LSPCleanUpTime + LSPChageOverTime) / 60 * BurdenedDIRLAB);


CostofAcquisitionFODSweeper = 3 * 7000;
(* Sweeper data is for the FOD*Boss in the Triplex Trailer Configuration *)
CostofVehicleUseperMile = 0.1375;
(* Operating costs of light truck per mile, 2014 data, 2015 is not available yet *)
TotalCostofVisualInspectionsandRunwaySweep = 0;
TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisition = 0;
Do[TotalCostofVisualInspectionsandRunwaySweep =
  2.704 * ((nRunwaySweeps * (Lrunway * 0.000621371) * (Wrunway / Wsweeper) * CostofVehicleUseperMile) +
      ((Lrunway * Wrunway / FODBossSweptAreaperhour) * nRunwaySweeps * 2 * BurdenedDIRLAB) +
      (Ninspections * TvisualInspection * InspectionVehicleSpeed) * CostofVehicleUseperMile +
      (Ninspections * TvisualInspection * 2) * BurdenedDIRLAB) +
   TotalCostofVisualInspectionsandRunwaySweep, {nn, 1, Ntakeoffs}];
TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisition =
 CostofAcquisitionFODSweeper / Naircraft + TotalCostofVisualInspectionsandRunwaySweep +
  CostofAcquisitionLSPMachine / Naircraft + LSPProcessCost;


Do[
 Print[Style["Takeoff Lifetime: ", 36, Red], n];
 Ncycles = 0;
 NcyclesNaturalGrowth = 0;
 NcyclesAppended = {};
 NcyclesAppendedTotal = {};
 NcyclesNaturalGrowthAppended = {};
 aaAppended = {};
 aaAppendedNaturalGrowth = {};
 Ncyclesbladereplaced = 0;
 Nbladereplacements = 0;
 NcyclesatImpact = {};
 CostIncreaseDuetoEngineEfficiencyDegradation = 0;


 LifeTimerepairs = Nrepairs;
 LifeTimereplacements = Nreplacements;
 LifeTimeEnginereplacements = NEnginereplacements;

 Do[
  Label[Restart];

  (* Probability of FOD under/within Vortex Core from Visual Inspections *)
  PoFodunderVortexCore =
   (1/Lrunway (-0.25` Lrunway + Wrunway (-0.43301270189221924` + 0.43301270189221924`
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
                                                
```

```
6 |
      (h - 0.285513648014925) * ((HeadWind - 10.6542844452363) * -0.0000000417742351687) +
      (GroundDistance - 0.727088406885572) * ((HeadWind - 10.6542844452363) * -0.0000000044377544096) +
      (InletMachNumber - 0.472574813666666) *
       ((HeadWind - 10.6542844452363) * -0.0000000678667866207) +
      (HeadWind - 10.6542844452363) * ((HeadWind - 10.6542844452363) * -0.0000000800667122655);

  Rimpact = 0.065415445705254 + -0.156714564501262 * h + 0.000132668082703971 * GroundDistance +
     0.64470128769476 * InletMachNumber + 0.00178339145613628 * HeadWind +
     (h - 0.285513648014925) * ((h - 0.285513648014925) * 0.00161247997422258) +
     (h - 0.285513648014925) * ((GroundDistance - 0.727088406885572) * 0.000201935239468427) +
     (GroundDistance - 0.727088406885572) *
      ((GroundDistance - 0.727088406885572) * -0.000362165459376822) +
     (h - 0.285513648014925) * ((InletMachNumber - 0.472574813666666) * 0.792303049785879) +
     (GroundDistance - 0.727088406885572) * ((InletMachNumber - 0.472574813666666) *
        0.00131731731841904) + (InletMachNumber - 0.472574813666666) *
      ((InletMachNumber - 0.472574813666666) * -0.00732966024459297) +
     (h - 0.285513648014925) * ((HeadWind - 10.6542844452363) * -0.0000312086878645552) +
     (GroundDistance - 0.727088406885572) * ((HeadWind - 10.6542844452363) * -0.00000324257013050 8) +
     (InletMachNumber - 0.472574813666666) *
      ((HeadWind - 10.6542844452363) * -0.0000516841213850941) +
     (HeadWind - 10.6542844452363) * ((HeadWind - 10.6542844452363) * -0.0000598168644780257);

  XpercentChordmin = 0;
XpercentChordmax = 0.94719;
  alphaXpercentChord = 0.8478;
  betaXpercentChord = 1.59718;

  XpercentChord = InverseBetaRegularized[RandomReal[], alphaXpercentChord, betaXpercentChord] *
      (XpercentChordmax - XpercentChordmin) + XpercentChordmin;

  DepthofPenetrationmin = 0.000029;
DepthofPenetrationmax = 0.00051;
  alphaDepthofPenetration = 1.4797852;
  betaDepthofPenetration = 3.5553198;

  LocalThicknessatImpact =
   0.00186625343042413 + 0.00132154137591184 * XpercentChord + (XpercentChord - 0.317678693072139) *
      ((XpercentChord - 0.317678693072139) * -0.0080516775782884) +
     (XpercentChord - 0.317678693072139) * ((XpercentChord - 0.317678693072139) *
        ((XpercentChord - 0.317678693072139) * 0.00548564206578457)) +
     (XpercentChord - 0.317678693072139) * ((XpercentChord - 0.317678693072139) * ((XpercentChord -
           0.317678693072139) * ((XpercentChord - 0.317678693072139) * -0.0402999953775497))) +
     (XpercentChord - 0.317678693072139) * ((XpercentChord - 0.317678693072139) *
        ((XpercentChord - 0.317678693072139) * ((XpercentChord - 0.317678693072139) *
           ((XpercentChord - 0.317678693072139) * 0.0530674615909787))));

  ImpactSide = If[RandomReal[] ≤ 283 / 402, "Pressure Side", "Suction Side"];

  EIFSa =0.00025106738269976316;
  EIFsDiam = 2 * EIFSa;
  aa = EIFSa;

Printed by Wolfram Mathematica Student Edition
```

Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
 Diameter2c = EIFsDiam;
 aa0 = EIFSa;
 aaNaturalGrowth = EIFSa;
 Diameter2cNaturalGrowth = 2 * EIFSa;

 BetaCompoundedAppend = {};
 BetaCompoundedAppendNaturalGrowth = {};
 Impact1 = 0;
 ChordatImpact0 = ChordatImpact;
 XpercentChord0 = XpercentChord; (*XpercentChord;*)
 DeltaSigma = 2 * Axialσ * (1 - R) / (1 + R);
 ImpactEccentricty = ChordatImpact0 * Abs[XpercentChord0 - 0.5];
 Print["ImpactEccentricty: ", ImpactEccentricty],
 Axialσ = 0;
 ChordatImpact = 0;
 XpercentChord = 0;
 DepthofPenetration = 0;
 LocalThicknessatImpact = 0;
];

If[ChordatImpact == 0, i = i + 1;
 If[i == Ntakeoffs, II = 1;
  Print["No Particle Impacts During Life of Blade"];
  Goto[End2], Goto[Restart]], If[i == Ntakeoffs, Goto[End1]]];

If[0.05 ≤ XpercentChord ≤ 0.80, AverageAirfoilThickness = LocalThicknessatImpact,
 AverageAirfoilThickness = 0.0551 * ChordatImpact0];
MaxAirfoilThickness = 0.10 * ChordatImpact;
Kic = 222.794957946974 +
   -9579.98234775552 * MaxAirfoilThickness + (MaxAirfoilThickness - 0.0079022222222222) *
    ((MaxAirfoilThickness - 0.0079022222222222) * 164 325.655251741) +
   (MaxAirfoilThickness - 0.0079022222222222) * ((MaxAirfoilThickness - 0.0079022222222222) *
      ((MaxAirfoilThickness - 0.0079022222222222) * 89 110 035.2890032)) +
   (MaxAirfoilThickness - 0.0079022222222222) * ((MaxAirfoilThickness - 0.0079022222222222) *
      ((MaxAirfoilThickness - 0.0079022222222222) *
        ((MaxAirfoilThickness - 0.0079022222222222) * -34 579 079 833.1077))) +
   (MaxAirfoilThickness - 0.0079022222222222) * ((MaxAirfoilThickness - 0.0079022222222222) *
      ((MaxAirfoilThickness - 0.0079022222222222) * ((MaxAirfoilThickness - 0.0079022222222222) *
          ((MaxAirfoilThickness - 0.0079022222222222) * 2 523 050 943 520.48))));

If[ChordatImpact > 0,
 Impact1 = 1;
 Print["Takeoff When Impact Occurs: ", Impact1];
 Print["XpercentChord: ", XpercentChord];

 Nimpacts = Nimpacts + 1;
 NimpactsAppended = AppendTo[NimpactsAppended, Nimpacts];

 AxialσMax = 2 * Axialσ / (1 + R);
 KMaxatFracture = Kic;
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
8 |
If[(0.80 ≥ XpercentChord ≥ 0.05) && ChordatImpact > 0,

 aac50thpercentilethumbnailflaw=0.000650084093009177;
 aac = aac50thpercentilethumbnailflaw;

 (* Clear[aac];
 aac=aac/.Extract[
    FindMinimum[{{{1.12*(1.08882991825178+0.191072321954091*(aac/AverageAirfoilThickness)+
              -0.2275*(aac/(2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                   ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5]))+
              ((aac/AverageAirfoilThickness)-0.2375)*(((aac/AverageAirfoilThickness)-
                   0.2375)*-0.336251716696097)+((aac/AverageAirfoilThickness)-0.2375)*
               (((aac/(2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                        ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^
                         0.5]))-0.256666666666667)*-0.642299410757611)+
              ((aac/(2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                     ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5]))-0.256666666666667)*
               (((aac/(2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                        ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5]))-0.256666666666667)*
                 0.443509615384615))*(Pi*(Log[(ChordatImpact/2+ImpactEccentricty+
                   ((ChordatImpact/2+ImpactEccentricty)^2-
                      ((2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                           ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)^2)^0.5)/
                  ((2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                        ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)])/
               ((ArcCos[(2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                        ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/
                     (ChordatImpact-2*ImpactEccentricty)])*(Log[(ChordatImpact/2+
                      ImpactEccentricty+((ChordatImpact/2+ImpactEccentricty)^2-
                         ((2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                              ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)^2)^0.5)/
                     ((2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                           ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)])+
                  (ArcCos[(2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^
                           0.5, ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/
                     (ChordatImpact+2*ImpactEccentricty)])*(Log[(ChordatImpact/2-
                      ImpactEccentricty+((ChordatImpact/2-ImpactEccentricty)^2-
                         ((2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                              ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)^2)^0.5)/
                     ((2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                           ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2)]))*
               (((aac/((2*If[aac≤FodDiam/2, ((FodDiam/2)^2-(FodDiam/2-aac)^2)^0.5,
                        ((FodDiam/2)^2-(FodDiam/2-FodDiam/2)^2)^0.5])/2))^0.5)/
                 (1/4 π (1+(aac)^2/If[2 aac≤FodDiam, (aac (-aac+FodDiam))^0.5`, 0.5`
                      ((FodDiam)^2)^0.5`]^2)))*(Pi*aac)^0.5)-
            (KMaxatFracture)/((Axialσ/10^6))}^2, aac≥0}, {aac, 0.0001}], {2}];

 Print["aac: ",aac];

 aacAppendedThumbnailFlaw=AppendTo[aacAppendedThumbnailFlaw, aac];
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
Print["aacAppendedThumbnailFlaw: ",aacAppendedThumbnailFlaw]; *)


Do[
 aaAppendedNaturalGrowth = AppendTo[aaAppendedNaturalGrowth, aaNaturalGrowth];
 NcyclesNaturalGrowthAppended =
  AppendTo[NcyclesNaturalGrowthAppended, NcyclesNaturalGrowth];
 BetaCFC = ((aaNaturalGrowth / (Diameter2cNaturalGrowth / 2)) ^ 0.5) / NIntegrate[
    (1 - (1 - (aaNaturalGrowth / (Diameter2cNaturalGrowth / 2)) ^ 2) * (Sin[φ]) ^ 2), {φ, 0, Pi / 2}];
 BetaBFS = 1.12;
 G1 = ArcCos[Diameter2cNaturalGrowth / (ChordatImpact - 2 * ImpactEccentricty)];
 G2 = ArcCos[Diameter2cNaturalGrowth / (ChordatImpact + 2 * ImpactEccentricty)];
 G3 = Log[(ChordatImpact / 2 - ImpactEccentricty + ((ChordatImpact / 2 - ImpactEccentricty) ^
          2 - (Diameter2cNaturalGrowth / 2) ^ 2) ^ 0.5) / (Diameter2cNaturalGrowth / 2)];
 G4 = Log[(ChordatImpact / 2 + ImpactEccentricty + ((ChordatImpact / 2 + ImpactEccentricty) ^
          2 - (Diameter2cNaturalGrowth / 2) ^ 2) ^ 0.5) / (Diameter2cNaturalGrowth / 2)];
 BetaW = Pi * G4 / (G1 * G4 + G2 * G3);
 BetaFFS = 1.08882991825178 + 0.191072321954091 * (aaNaturalGrowth / AverageAirfoilThickness) +
   -0.2275 * (aaNaturalGrowth / Diameter2cNaturalGrowth) +
   ((aaNaturalGrowth / AverageAirfoilThickness) - 0.2375) *
    (((aaNaturalGrowth / AverageAirfoilThickness) - 0.2375) * -0.336251716696097) +
   ((aaNaturalGrowth / AverageAirfoilThickness) - 0.2375) *
    (((aaNaturalGrowth / Diameter2cNaturalGrowth) - 0.256666666666667) * -0.642299410757611) +
   ((aaNaturalGrowth / Diameter2cNaturalGrowth) - 0.256666666666667) *
    (((aaNaturalGrowth / Diameter2cNaturalGrowth) - 0.256666666666667) * 0.443509615384615);
 BetaCompounded = BetaBFS * BetaFFS * BetaW * BetaCFC;
 BetaCompoundedAppendNaturalGrowth =
  AppendTo[BetaCompoundedAppendNaturalGrowth, BetaCompounded];
 DeltaK = BetaCompounded * DeltaSigma * (Pi * aaNaturalGrowth) ^ 0.5;
 σResidual = 0;
 If[ImpactSide == "Pressure Side",
  If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 1, σResidual = -InverseCDF[
        NormalDistribution[Abs[(-70.7331771213364) + 117116.263407782 * aaNaturalGrowth +
            (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * -395474551.758495 +
            (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth -
                0.000335) * 600608204212.839 + (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth -
                0.000335) * (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
             -313392042189632], Abs[((-91.388930233299) + 163881.618156941 * aaNaturalGrowth +
              (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * -621710641.059832 +
              (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
               (aaNaturalGrowth - 0.000335) * 784096231132.582 + (aaNaturalGrowth - 0.000335) *
               (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
               (aaNaturalGrowth - 0.000335) * -181928735061177) - ((-51.0322827764947) +
             73621.0917592117 * aaNaturalGrowth + (aaNaturalGrowth - 0.000335) *
               (aaNaturalGrowth - 0.000335) * -186084270.781473 + (aaNaturalGrowth - 0.000335) *
               (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * 369136849298.813 +
              (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth -
                 0.000335) * (aaNaturalGrowth - 0.000335) * -331072585461178)] / 6],
        RandomReal[]] * If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1,
        If[LSPTreatment == 3, 3, If[LSPTreatment == 0, 0]]]]],
  If[ImpactSide == "Suction Side", If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 0.80,
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
10 |
        σResidual = -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) +
              117116.263407782 * aaNaturalGrowth + (aaNaturalGrowth - 0.000335) *
               (aaNaturalGrowth - 0.000335) * -395474551.758495 + (aaNaturalGrowth - 0.000335) *
               (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * 600608204212.839 +
              (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
               (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * -313392042189632],
            Abs[((-91.388930233299) + 163881.618156941 * aaNaturalGrowth +
                 (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * -621710641.059832 +
                 (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
                  (aaNaturalGrowth - 0.000335) * 784096231132.582 +
                 (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth -
                    0.000335) * (aaNaturalGrowth - 0.000335) * -181928735061177) -
               ((-51.0322827764947) + 73621.0917592117 * aaNaturalGrowth +
                 (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * -186084270.781473 +
                 (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
                  (aaNaturalGrowth - 0.000335) * 369136849298.813 + (aaNaturalGrowth -
                    0.000335) * (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
                  (aaNaturalGrowth - 0.000335) * -331072585461178)]/6], RandomReal[]] *
          If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1, If[LSPTreatment == 3,
             3, If[LSPTreatment == 0, 0]]]]]];

      Kmax = BetaCompounded * If[aaNaturalGrowth ≤ 0.00091,
          AxialσMax + σResidual * 10^6, AxialσMax] * (P1 * aaNaturalGrowth)^0.5;
      dadN = (0.0000000000052) * ((DeltaK/10^6)^2.5) * ((Kmax/10^6)^0.67);
      DeltaNcycles = Nincremont;
      daa = dadN * DeltaNcycles;
      aaNaturalGrowth = aaNaturalGrowth + daa;
      If[aaNaturalGrowth === ComplexInfinity || Head[aaNaturalGrowth] === Complex,
       aaNaturalGrowth = Re[aaNaturalGrowth]];
      Diameter2cNaturalGrowth = 2 * aaNaturalGrowth;
      NcyclesNaturalGrowth = ii * Nincremont;
      If[aaNaturalGrowth ≥ aac || Ntakeoffs * 2 * 60 * 60 * StressFrequency == Ncycles,
       Print["ith takeoff during Natural Growth when a >= aac or
           Ncycles = Ntakeoffs*2*60*60*StressFrequency : ", ii];
       NcyclesNaturalGrowth = 0; Goto[Next1]],
      {ii, 1, Ntakeoffs}];

    Label[Next1];


    Do[
      aaAppended = AppendTo[aaAppended, aa];
      NcylesAppended = AppendTo[NcylesAppended, Ncycles];
      BetaCFC = ((aa/(Diameter2c/2))^0.5)/
        NIntegrate[(1 - (1 - (aa/(Diameter2c/2))^2) * (Sin[φ])^2), {φ, 0, P1/2}];
      BetaBFS = 1.12;
      G1 = ArcCos[Diameter2c/(ChordatImpact - 2 * ImpactEccentricty)];
      G2 = ArcCos[Diameter2c/(ChordatImpact + 2 * ImpactEccentricty)];
      G3 = Log[(ChordatImpact/2 - ImpactEccentricty +
            ((ChordatImpact/2 - ImpactEccentricty)^2 - (Diameter2c/2)^2)^0.5)/(Diameter2c/2)];
      G4 = Log[(ChordatImpact/2 + ImpactEccentricty + ((ChordatImpact/2 + ImpactEccentricty)^
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
        2 - (Diameter2c/2)^2)^0.5) / (Diameter2c/2)];
BetaW = P1 * G4 / (G1 * G4 + G2 * G3);
BetaFFS = 1.08882991825178 + 0.191072321954091 * (aa / AverageAirfoilThickness) +
  -0.2275 * (aa / Diameter2c) + ((aa / AverageAirfoilThickness) - 0.2375) *
   (((aa / AverageAirfoilThickness) - 0.2375) * -0.336251716696097) +
  ((aa / AverageAirfoilThickness) - 0.2375) * (((aa / Diameter2c) - 0.256666666666667) *
     -0.642299410757611) + ((aa / Diameter2c) - 0.256666666666667) *
   (((aa / Diameter2c) - 0.256666666666667) * 0.443509615384615);
BetaCompounded = BetaBFS * BetaFFS * BetaW * BetaCFC;
BetaCompoundedAppend = AppendTo[BetaCompoundedAppend, BetaCompounded];
DeltaK = BetaCompounded * DeltaSigma * (P1 * aa)^0.5;

σResidual = 0;
If[ImpactSide == "Pressure Side",
 If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 1, σResidual = -InverseCDF[NormalDistribution[
       Abs[(-70.7331771213364) + 117116.263407782 * aa + (aa - 0.000335) * (aa - 0.000335) *
          -395474551.758495 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
          600608204212.839 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
          (aa - 0.000335) * -313392042189632], Abs[((-91.388930233299) + 163881.618156941 *
             aa + (aa - 0.000335) * (aa - 0.000335) * -621710641.059832 + (aa - 0.000335) *
             (aa - 0.000335) * (aa - 0.000335) * 784096231132.582 + (aa - 0.000335) *
             (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -181928735061177) -
          ((-51.0322827764947) + 73621.0917592117 * aa + (aa - 0.000335) * (aa - 0.000335) *
             -186084270.781473 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
             369136849298.813 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
             (aa - 0.000335) * -331072585461178)]/6], RandomReal[]] * If[LSPTreatment == 1,
    0.75, If[LSPTreatment == 2, 1, If[LSPTreatment == 3, 3, If[LSPTreatment == 0, 0]]]]],
 If[ImpactSide == "Suction Side", If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 0.80,
    σResidual = -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) + 117116.263407782 *
          aa + (aa - 0.000335) * (aa - 0.000335) * -395474551.758495 + (aa - 0.000335) *
          (aa - 0.000335) * (aa - 0.000335) * 600608204212.839 + (aa - 0.000335) *
          (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -313392042189632],
        Abs[((-91.388930233299) + 163881.618156941 * aa + (aa - 0.000335) * (aa - 0.000335) *
             -621710641.059832 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
             784096231132.582 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
             (aa - 0.000335) * -181928735061177) - ((-51.0322827764947) + 73621.0917592117 *
             aa + (aa - 0.000335) * (aa - 0.000335) * -186084270.781473 + (aa - 0.000335) *
             (aa - 0.000335) * (aa - 0.000335) * 369136849298.813 + (aa - 0.000335) *
             (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -331072585461178)]/6],
       RandomReal[]] * If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1, If[
        LSPTreatment == 3, 3, If[LSPTreatment == 0, 0]]]]]];

Kmax = BetaCompounded * If[aa ≤ 0.00091, AxialσMax + σResidual * 10^6, AxialσMax] * (P1 * aa)^0.5;
dadN = (0.0000000000052) * ((DeltaK/10^6)^2.5) * ((Kmax/10^6)^0.67);
DeltaNcycles = Nincrement;
daa = dadN * DeltaNcycles;
aa = aa + daa;
If[aa === ComplexInfinity || Head[aa] === Complex, aa = Re[aa]];
Diameter2c = 2 * aa;
Ncycles = Ncycles + Nincrement;
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
12 |
        LevelIIIDamageRandomSeed = RandomReal[];
        If[EngineIntervalInspections == 1,
         If[Ncycles == NcyclesIntervalInspection && aaD ≤ aa < aac, Nrepairs = Nrepairs + 1;
          aa = EIFSa, If[LevelIIIDamageRandomSeed ≥ 4 * 10^-5,
           If[Ncycles == NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
            aa = EIFSa], If[LevelIIIDamageRandomSeed < 4 * 10^-5 && Ncycles ==
              NcyclesIntervalInspection && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
            aa = EIFSa;
            Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]]];
         If[LevelIIIDamageRandomSeed > 4 * 10^-5,
          If[Ncycles < NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
           aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles <
              NcyclesIntervalInspection && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
            aa = EIFSa;
            Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]]];
         If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[NcyclesIntervalInspection * 1.5 > Ncycles >
             NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
           aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5,
           If[NcyclesIntervalInspection * 1.5 > Ncycles > NcyclesIntervalInspection && aa ≥ aac,
            NEngineReplacements = NEngineReplacements + 1;
            aa = EIFSa;
            Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]]];
         If[Ncycles == NcyclesIntervalInspection * 1.5 && aaD ≤ aa < aac, Nrepairs = Nrepairs + 1;
          aa = EIFSa, If[LevelIIIDamageRandomSeed ≥ 4 * 10^-5,
           If[Ncycles == NcyclesIntervalInspection * 1.5 && aa ≥ aac, Nreplacements = Nreplacements + 1;
            aa = EIFSa], If[LevelIIIDamageRandomSeed < 4 * 10^-5 && Ncycles ==
              NcyclesIntervalInspection * 1.5 && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
            aa = EIFSa;
            Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]]];
         If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[Ntakeoffs * 2 * 60 * 60 * StressFrequency > Ncycles >
             NcyclesIntervalInspection * 1.5 && aa ≥ aac, Nreplacements = Nreplacements + 1;
           aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles >
              NcyclesIntervalInspection * 1.5 && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
            aa = EIFSa;
            Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]],
         If[EngineIntervalInspections == 0, If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[Ncycles <
              Ntakeoffs * 2 * 60 * 60 * StressFrequency && aa ≥ aac, Nreplacements = Nreplacements + 1;
            aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles < Ntakeoffs * 2 *
                60 * 60 * StressFrequency && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
             aa = EIFSa;
             Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]]]],
        {ii, 1, Impacti}];


Print["AxialσMax: ", AxialσMax];
Print["σResidual*10^6: ", σResidual * 10^6];

Print["Nrepairs: ", Nrepairs];
Print["Nreplacements: ", Nreplacements];
Print["NEngineReplacements: ", NEngineReplacements];
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
If[aa ≤ DepthofPenetration, aa = DepthofPenetration;
 Diameter2c = 2 * If[aa ≤ FodDiam/2, ((FodDiam/2)^2 - (FodDiam/2 - aa)^2)^0.5,
    ((FodDiam/2)^2 - (FodDiam/2 - FodDiam/2)^2)^0.5], Diameter2c = 2 * aa];


(* aa0regressed=aa;
Do[
 BetaCFC=((aa0regressed/(Diameter2c/2))^0.5)/
   NIntegrate[(1-(1-(aa0regressed/(Diameter2c/2))^2)*(Sin[φ])^2),{φ,0,P1/2}];
 BetaBFS=1.12;
 G1=ArcCos[Diameter2c/(ChordatImpact-2*ImpactEccentricty)];
 G2=ArcCos[Diameter2c/(ChordatImpact+2*ImpactEccentricty)];
 G3=Log[(ChordatImpact/2-ImpactEccentricty+
      ((ChordatImpact/2-ImpactEccentricty)^2-(Diameter2c/2)^2)^0.5)/(Diameter2c/2)];
 G4=Log[(ChordatImpact/2+ImpactEccentricty+((ChordatImpact/2+ImpactEccentricty)^
         2-(Diameter2c/2)^2)^0.5)/(Diameter2c/2)];
 BetaW=P1*G4/(G1*G4+G2*G3);
 BetaFFS=1.08882991825178+0.191072321954091*(aa0regressed/AverageAirfoilThickness)+
   -0.2275*(aa0regressed/Diameter2c)+((aa0regressed/AverageAirfoilThickness)-0.2375)*
    (((aa0regressed/AverageAirfoilThickness)-0.2375)*-0.336251716696097)+
   ((aa0regressed/AverageAirfoilThickness)-0.2375)*
    (((aa0regressed/Diameter2c)-0.256666666666667)*-0.642299410757611)+
   ((aa0regressed/Diameter2c)-0.256666666666667)*
    (((aa0regressed/Diameter2c)-0.256666666666667)*0.443509615384615);
 BetaCompounded=BetaBFS*BetaFFS*BetaW*BetaCFC;
 BetaCompoundedAppend=AppendTo[BetaCompoundedAppend,BetaCompounded];
 DeltaK=BetaCompounded*DeltaSigma*(P1*aa0regressed)^0.5;
 Kmax=BetaCompounded*AxialcMax*(P1*aa0regressed)^0.5;
 aa0regressed=
  aa0regressed-(0.0000000000052)*((DeltaK/10^6)^2.5)*((Kmax/10^6)^0.67)*Nincrement,
 {ii,1,Impacti+1)];


Print["aa0regressed: ",aa0regressed," aa at Impact: ",aa];
If[aa0regressed===ComplexInfinity||Head[aa0regressed]===Complex,
 Goto[Next5],aa0regressedAppended=AppendTo[aa0regressedAppended,aa0regressed]];
Label[Next5]; *)

aaAppended = AppendTo[aaAppended, aa];
NcyclesAppended = AppendTo[NcyclesAppended, Ncycles];
Print["Ncycles Between Initial Growth and Impact: ", Ncycles];
Print["NcyclesatImpact: ", Impacti * Nincrement];

Do[
 BetaCFC = ((aa/(Diameter2c/2))^0.5)/
   NIntegrate[(1-(1-(aa/(Diameter2c/2))^2)*(Sin[φ])^2), {φ, 0, P1/2}];
 BetaBFS = 1.12;
 G1 = ArcCos[Diameter2c/(ChordatImpact-2*ImpactEccentricty)];
 G2 = ArcCos[Diameter2c/(ChordatImpact+2*ImpactEccentricty)];
 G3 = Log[(ChordatImpact/2-ImpactEccentricty+
      ((ChordatImpact/2-ImpactEccentricty)^2-(Diameter2c/2)^2)^0.5)/(Diameter2c/2)];
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
14 |
G4 = Log[(ChordatImpact / 2 + ImpactEccentricty + ((ChordatImpact / 2 + ImpactEccentricty) ^
          2 - (Diameter2c / 2) ^ 2) ^ 0.5) / (Diameter2c / 2)];
BetaW = P1 * G4 / (G1 * G4 + G2 * G3);
BetaFFS = 1.08882991825178 + 0.191072321954091 * (aa / AverageAirfoilThickness) +
  -0.2275 * (aa / Diameter2c) + ((aa / AverageAirfoilThickness) - 0.2375) *
   (((aa / AverageAirfoilThickness) - 0.2375) * -0.336251716696097) +
  ((aa / AverageAirfoilThickness) - 0.2375) * (((aa / Diameter2c) - 0.256666666666667) *
     -0.642299410757611) + ((aa / Diameter2c) - 0.256666666666667) *
   (((aa / Diameter2c) - 0.256666666666667) * 0.443509615384615);
BetaCompounded = BetaBFS * BetaFFS * BetaW * BetaCFC;
BetaCompoundedAppend = AppendTo[BetaCompoundedAppend, BetaCompounded];
DeltaK = BetaCompounded * DeltaSigma * (P1 * aa) ^ 0.5;

σResidual = 0;
If[ImpactSide == "Pressure Side",
 If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 1, σResidual = -InverseCDF[NormalDistribution[
        Abs[(-70.7331771213364) + 117116.263407782 * aa + (aa - 0.000335) * (aa - 0.000335) *
           -395474551.758495 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
           600608204212.839 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
           (aa - 0.000335) * -313392042189632], Abs[((-91.388930233299) + 163881.618156941 *
              aa + (aa - 0.000335) * (aa - 0.000335) * -621710641.059832 + (aa - 0.000335) *
              (aa - 0.000335) * (aa - 0.000335) * 784096231132.582 + (aa - 0.000335) *
              (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -181928735061177) -
           ((-51.0322827764947) + 73621.0917592117 * aa + (aa - 0.000335) * (aa - 0.000335) *
              -186084270.781473 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
              369136849298.813 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
              (aa - 0.000335) * -331072585461178)] / 6], RandomReal[] * If[LSPTreatment == 1,
      0.75, If[LSPTreatment == 2, 1, If[LSPTreatment == 3, 3, If[LSPTreatment == 0, 0]]]]],
 If[ImpactSide == "Suction Side", If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 0.80,
    σResidual = -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) + 117116.263407782 *
           aa + (aa - 0.000335) * (aa - 0.000335) * -395474551.758495 + (aa - 0.000335) *
           (aa - 0.000335) * (aa - 0.000335) * 600608204212.839 + (aa - 0.000335) *
           (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -313392042189632],
         Abs[((-91.388930233299) + 163881.618156941 * aa + (aa - 0.000335) * (aa - 0.000335) *
              -621710641.059832 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
              784096231132.582 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
              (aa - 0.000335) * -181928735061177) - ((-51.0322827764947) + 73621.0917592117 *
              aa + (aa - 0.000335) * (aa - 0.000335) * -186084270.781473 + (aa - 0.000335) *
              (aa - 0.000335) * (aa - 0.000335) * 369136849298.813 + (aa - 0.000335) *
              (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -331072585461178)] / 6],
       RandomReal[] * If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1, If[
        LSPTreatment == 3, 3, If[LSPTreatment == 0, 0]]]]]]];

Kmax = BetaCompounded * If[aa ≤ 0.00091, AxialσMax + σResidual * 10^6, AxialσMax] * (P1 * aa) ^ 0.5;
dadN = (0.0000000000052) * ((DeltaK / 10^6) ^ 2.5) * ((Kmax / 10^6) ^ 0.67);
DeltaNcycles = Nincrement;
daa = dadN * DeltaNcycles;
aa = aa + daa;
If[aa === ComplexInfinity || Head[aa] === Complex, aa = Re[aa]];
If[aa ≤ DepthofPenetration,
 Diameter2c = 2 * If[aa ≤ Diameter2c / 2, ((Diameter2c / 2) ^ 2 - (Diameter2c / 2 - aa) ^ 2) ^ 0.5,

Printed by Wolfram Mathematica Student Edition
```

Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
    ((Diameter2c/2)^2 - (Diameter2c/2 - Diameter2c/2)^2)^0.5], Diameter2c = 2 * aa];
Ncycles = 11 * Nincrement;
nrepairs = 0;
nreplacements = 0;
LevelIIIDamageRandomSeed = RandomReal[];
If[EngineIntervalInspections == 1,
 If[Ncycles == NcyclesIntervalInspection && aaD ≤ aa < aac, Nrepairs = Nrepairs + 1;
  aa = EIFSa, If[LevelIIIDamageRandomSeed ≥ 4 * 10^-5,
   If[Ncycles == NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
    aa = EIFSa], If[LevelIIIDamageRandomSeed < 4 * 10^-5 && Ncycles ==
      NcyclesIntervalInspection && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]]];
 If[LevelIIIDamageRandomSeed > 4 * 10^-5,
  If[Ncycles < NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
   aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles <
      NcyclesIntervalInspection && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]]];
 If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[NcyclesIntervalInspection * 1.5 > Ncycles >
     NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
   aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5,
   If[NcyclesIntervalInspection * 1.5 > Ncycles > NcyclesIntervalInspection && aa ≥ aac,
    NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]]];
 If[Ncycles == NcyclesIntervalInspection * 1.5 && aaD ≤ aa < aac, Nrepairs = Nrepairs + 1;
  aa = EIFSa, If[LevelIIIDamageRandomSeed ≥ 4 * 10^-5,
   If[Ncycles == NcyclesIntervalInspection * 1.5 && aa ≥ aac, Nreplacements = Nreplacements + 1;
    aa = EIFSa], If[LevelIIIDamageRandomSeed < 4 * 10^-5 && Ncycles ==
      NcyclesIntervalInspection * 1.5 && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]]];
 If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[Ntakeoffs * 2 * 60 * 60 * StressFrequency > Ncycles >
     NcyclesIntervalInspection * 1.5 && aa ≥ aac, Nreplacements = Nreplacements + 1;
   aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles >
      NcyclesIntervalInspection * 1.5 && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]],
 If[EngineIntervalInspections == 0, If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[Ncycles <
      Ntakeoffs * 2 * 60 * 60 * StressFrequency && aa ≥ aac, Nreplacements = Nreplacements + 1;
    aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles < Ntakeoffs * 2 *
        60 * 60 * StressFrequency && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
     aa = EIFSa;
     Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]]]];

If[aac ≤ aa || Ntakeoffs * 2 * 60 * 60 * StressFrequency == Ncycles,
 Print["ith takeoff during Natural Growth when a >= aac or
    Ncycles = Ntakeoffs*2*60*60*StressFrequency: ", ii];
 i = i + 1;
 NcyclesatbladeReplacement = AppendTo[NcyclesatbladeReplacement, Ncycles];
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
16 |
    NcyclesatbladereplacementTotal = AppendTo[NcyclesatbladereplacementTotal, Ncycles];
    Nbladereplacements = Nbladereplacements + 1;
    Ncycles = 0,
    aaAppended = AppendTo[aaAppended, aa];
    NcyclesAppended = AppendTo[NcyclesAppended, Ncycles];
    NcyclesAppendedTotal =
     AppendTo[NcyclesAppendedTotal, Transpose[{NcyclesAppended, aaAppended}]];
   ],
   {ii, Impacti + 1, Ntakeoffs}];

If[Nrepairs > 0, CostIncreaseDuetoEngineEfficiencyDregradation =
   (0.005 * RandomReal[] * 0.005) * 2 * CostofFuelperFlight * (Ntakeoffs - Impacti)];
Print["CostIncreaseDuetoEngineEfficiencyDregradation: ",
 CostIncreaseDuetoEngineEfficiencyDregradation];


Print["Nrepairs: ", Nrepairs];
Print["Nreplacements: ", Nreplacements];
Print["NEnginereplacements: ", NEnginereplacements],

If[(XpercentChord > 0.80 || XpercentChord < 0.05) && ChordatImpact > 0,
 Print["Check that Edge Crack Beta Solution is triggered"];

 Print["ChordatImpact: ", ChordatImpact];
 Print["KMaxatFracture: ", KMaxatFracture];
 Print["Axialσ: ", Axialσ];

 Clear[aac];

 (* aac=(aac/.Extract[FindRoot[
       {(1.12-0.231*(1000000*aac/ChordatImpact)+10.550*(1000000*aac/ChordatImpact)^2-
            21.710*(1000000*aac/ChordatImpact)^3+30.382*(1000000*aac/ChordatImpact)^4)*
          (1000000*Pi*aac)^0.5-(KMaxatFracture)/((Axialσ/10^6))}, {aac, 0.1}], 1])*1000000;

 aacAppendedThroughFlaw=AppendTo[aacAppendedThroughFlaw, aac];
 Print["aacAppendedThroughFlaw: ",aacAppendedThroughFlaw]; *)

 aac50thpercentilethroughflaw =0.018372160295978216;
 aac = aac50thpercentilethroughflaw;

 Print["aac: ", aac];

 Do[
  BetaCompounded =
   1.12 - 0.231 * (aaNaturalGrowth / ChordatImpact) + 10.550 * (aaNaturalGrowth / ChordatImpact) ^ 2 -
    21.710 * (aaNaturalGrowth / ChordatImpact) ^ 3 + 30.382 * (aaNaturalGrowth / ChordatImpact) ^ 4;
  BetaCompoundedAppendNaturalGrowth = AppendTo[BetaCompoundedAppendNaturalGrowth,
    BetaCompounded];
  DeltaK = BetaCompounded * DeltaSigma * (Pi * aaNaturalGrowth) ^ 0.5;

  σResidual = 0;
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
If[ImpactSide == "Pressure Side", If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 1,
   σResidual = -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) +
         117 116.263407782 * aaNaturalGrowth + (aaNaturalGrowth - 0.000335) *
          (aaNaturalGrowth - 0.000335) * -395 474 551.758495 + (aaNaturalGrowth - 0.000335) *
          (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * 600 608 204 212.839 +
         (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
          (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * -313 392 042 189 632],
       Abs[((-91.388930233299) + 163 881.618156941 * aaNaturalGrowth +
          (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * -621 710 641.059832 +
          (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
           (aaNaturalGrowth - 0.000335) * 784 096 231 132.582 +
          (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth -
             0.000335) * (aaNaturalGrowth - 0.000335) * -181 928 735 061 177) -
         ((-51.0322827764947) + 73 621.0917592117 * aaNaturalGrowth +
          (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * -186 084 270.781473 +
          (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
           (aaNaturalGrowth - 0.000335) * 369 136 849 298.813 + (aaNaturalGrowth -
             0.000335) * (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
           (aaNaturalGrowth - 0.000335) * -331 072 585 461 178)] / 6], RandomReal[]] *
     If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1, If[LSPTreatment == 3,
       3, If[LSPTreatment == 0, 0]]]]],
 If[ImpactSide == "Suction Side", If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 0.80,
   σResidual = -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) +
         117 116.263407782 * aaNaturalGrowth + (aaNaturalGrowth - 0.000335) *
          (aaNaturalGrowth - 0.000335) * -395 474 551.758495 + (aaNaturalGrowth - 0.000335) *
          (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * 600 608 204 212.839 +
         (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
          (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * -313 392 042 189 632],
       Abs[((-91.388930233299) + 163 881.618156941 * aaNaturalGrowth +
          (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
           -621 710 641.059832 + (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth -
             0.000335) * (aaNaturalGrowth - 0.000335) * 784 096 231 132.582 +
          (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
           (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
           -181 928 735 061 177) - ((-51.0322827764947) + 73 621.0917592117 *
          aaNaturalGrowth + (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) *
           -186 084 270.781473 + (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth -
             0.000335) * (aaNaturalGrowth - 0.000335) * 369 136 849 298.813 +
          (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth - 0.000335) * (aaNaturalGrowth -
             0.000335) * (aaNaturalGrowth - 0.000335) * -331 072 585 461 178)] / 6],
       RandomReal[]] * If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1,
       If[LSPTreatment == 3, 3, If[LSPTreatment == 0, 0]]]]]];

Kmax = BetaCompounded * If[aaNaturalGrowth ≤ 0.00091,
   AxialσMax + σResidual * 10^6, AxialσMax] * (PI * aaNaturalGrowth)^0.5;
dadN = (0.0000000000052) * ((DeltaK / 10^6)^2.5) * ((Kmax / 10^6)^0.67);
DeltaNcycles = Nincrement;
daa = dadN * DeltaNcycles;
aaNaturalGrowth = aaNaturalGrowth + daa;
If[aaNaturalGrowth === ComplexInfinity || Head[aaNaturalGrowth] === Complex,
 aaNaturalGrowth = Re[aaNaturalGrowth]];
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
18 |
 Diameter2cNaturalGrowth = 2 * aaNaturalGrowth;
 NcyclesNaturalGrowth = 11 * Nincrement;
 aaAppendedNaturalGrowth = AppendTo[aaAppendedNaturalGrowth, aaNaturalGrowth];
 NcyclesNaturalGrowthAppended =
  AppendTo[NcyclesNaturalGrowthAppended, NcyclesNaturalGrowth];
 If[aaNaturalGrowth ≥ aac || Ntakeoffs * 2 * 60 * 60 * StressFrequency == Ncycles,
  Print["ith takeoff during Natural Growth when a >= aac or
     Ncycles = Ntakeoffs*2*60*60*StressFrequency: ", 11];
  Goto[Next3]],
 {11, 1, Ntakeoffs}];
Label[Next3];


Do[
 BetaCompounded = 1.12 - 0.231 * (aa / ChordatImpact) + 10.550 * (aa / ChordatImpact) ^ 2 -
   21.710 * (aa / ChordatImpact) ^ 3 + 30.382 * (aa / ChordatImpact) ^ 4;
 BetaCompoundedAppend = AppendTo[BetaCompoundedAppend, BetaCompounded];
 DeltaK = BetaCompounded * DeltaSigma * (Pi * aa) ^ 0.5;

 σResidual = 0;
 If[ImpactSide == "Pressure Side",
  If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 1, σResidual =
     -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) + 117 116.263407782 * aa + (aa -
              0.000335) * (aa - 0.000335) * -395 474 551.758495 + (aa - 0.000335) * (aa - 0.000335) *
            (aa - 0.000335) * 600 608 204 212.839 + (aa - 0.000335) * (aa - 0.000335) *
            (aa - 0.000335) * (aa - 0.000335) * -313 392 042 189 632],
        Abs[((-91.388930233299) + 163 881.618156941 * aa + (aa - 0.000335) *
             (aa - 0.000335) * -621 710 641.059832 + (aa - 0.000335) * (aa - 0.000335) *
             (aa - 0.000335) * 784 096 231 132.582 + (aa - 0.000335) * (aa - 0.000335) *
             (aa - 0.000335) * (aa - 0.000335) * -181 928 735 061 177) - ((-51.0322827764947) +
           73 621.0917592117 * aa + (aa - 0.000335) * (aa - 0.000335) * -186 084 270.781473 +
           (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * 369 136 849 298.813 +
           (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
            -331 072 585 461 178)] / 6], RandomReal[]] * If[LSPTreatment == 1, 0.75,
    If[LSPTreatment == 2, 1, If[LSPTreatment == 3, 3, If[LSPTreatment == 0, 0]]]]],
  If[ImpactSide == "Suction Side", If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 0.80,
    σResidual = -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) + 117 116.263407782 *
          aa + (aa - 0.000335) * (aa - 0.000335) * -395 474 551.758495 + (aa - 0.000335) *
          (aa - 0.000335) * (aa - 0.000335) * 600 608 204 212.839 + (aa - 0.000335) *
          (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -313 392 042 189 632],
       Abs[((-91.388930233299) + 163 881.618156941 * aa + (aa - 0.000335) * (aa - 0.000335) *
            -621 710 641.059832 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
            784 096 231 132.582 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * (aa -
              0.000335) * -181 928 735 061 177) - ((-51.0322827764947) + 73 621.0917592117 *
           aa + (aa - 0.000335) * (aa - 0.000335) * -186 084 270.781473 + (aa - 0.000335) *
           (aa - 0.000335) * (aa - 0.000335) * 369 136 849 298.813 + (aa - 0.000335) *
           (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -331 072 585 461 178)] / 6],
       RandomReal[]] * If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1,
       If[LSPTreatment == 3, 3, If[LSPTreatment == 0, 0]]]]]];

 Kmax = BetaCompounded * If[aa ≤ 0.00091, AxialσMax + σResidual * 10^6, AxialσMax] * (Pi * aa) ^ 0.5;
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
dadN = (0.0000000000052) * ((DeltaK/10^6)^2.5) * ((Kmax/10^6)^0.67);
DeltaNcycles = Nincrement;
daa = dadN * DeltaNcycles;
aa = aa + daa;
If[aa === ComplexInfinity || Head[aa] === Complex, aa = Re[aa]];
Diameter2c = 2 * aa;
Ncycles = 11 * Nincrement;
aaAppended = AppendTo[aaAppended, aa];
NcylesAppended = AppendTo[NcylesAppended, Ncycles];
LevelIIIDamageRandomSeed = RandomReal[];
If[EngineIntervalInspections == 1,
 If[Ncycles == NcyclesIntervalInspection && aaD ≤ aa < aac, Nrepairs = Nrepairs + 1;
  aa = EIFSa, If[LevelIIIDamageRandomSeed ≥ 4 * 10^-5,
   If[Ncycles == NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
    aa = EIFSa], If[LevelIIIDamageRandomSeed < 4 * 10^-5 && Ncycles ==
      NcyclesIntervalInspection && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
 If[LevelIIIDamageRandomSeed > 4 * 10^-5,
  If[Ncycles < NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
   aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles <
     NcyclesIntervalInspection && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
 If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[NcyclesIntervalInspection * 1.5 > Ncycles >
    NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
   aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5,
   If[NcyclesIntervalInspection * 1.5 > Ncycles > NcyclesIntervalInspection && aa ≥ aac,
    NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
 If[Ncycles == NcyclesIntervalInspection * 1.5 && aaD ≤ aa < aac, Nrepairs = Nrepairs + 1;
  aa = EIFSa, If[LevelIIIDamageRandomSeed ≥ 4 * 10^-5,
   If[Ncycles == NcyclesIntervalInspection * 1.5 && aa ≥ aac, Nreplacements = Nreplacements + 1;
    aa = EIFSa], If[LevelIIIDamageRandomSeed < 4 * 10^-5 &&
     Ncycles == NcyclesIntervalInspection * 1.5 && aa ≥ aac,
    NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
 If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[Ntakeoffs * 2 * 60 * 60 * StressFrequency > Ncycles >
    NcyclesIntervalInspection * 1.5 && aa ≥ aac, Nreplacements = Nreplacements + 1;
   aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5,
   If[Ncycles > NcyclesIntervalInspection * 1.5 && aa ≥ aac,
    NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]],
 If[EngineIntervalInspections == 0, If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[Ncycles <
     Ntakeoffs * 2 * 60 * 60 * StressFrequency && aa ≥ aac, Nreplacements = Nreplacements + 1;
    aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles < Ntakeoffs * 2 *
       60 * 60 * StressFrequency && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
     aa = EIFSa;
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
20 |
          Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]]]],
 {i1, 1, Impact1}];


Print["BetaCompounded: ", BetaCompounded];
Print["aa Check: ", aa];

If[aa ≤ DepthofPenetration, aa = DepthofPenetration;
 Diameter2c = 2 * If[aa ≤ FodDiam/2, ((FodDiam/2)^2 - (FodDiam/2 - aa)^2)^0.5,
    ((FodDiam/2)^2 - (FodDiam/2 - FodDiam/2)^2)^0.5], Diameter2c = 2 * aa];

(* aa0regressed=aa;
Print["Regressed aa before Do Loop starts:",aa0regressed];

Do[
 BetaCompounded=
  1.12-0.231*(aa0regressed/ChordatImpact)+10.550*(aa0regressed/ChordatImpact)^2-
   21.710*(aa0regressed/ChordatImpact)^3+30.382*(aa0regressed/ChordatImpact)^4;
 DeltaK=BetaCompounded*DeltaSigma*(Pi*aa0regressed)^0.5;
 Kmax=BetaCompounded*AxialσMax*(Pi*aa0regressed)^0.5;
 aa0regressed=
  aa0regressed-(0.0000000000052)*((DeltaK/10^6)^2.5)*((Kmax/10^6)^0.67)*Nincrement,
 {i1,1,Impact1+1}];

Print["aa0regressed: ",aa0regressed," aa at Impact: ",aa];

If[aa0regressed===ComplexInfinity||Head[aa0regressed]===Complex,
 Goto[Next2],aa0regressedAppended=AppendTo[aa0regressedAppended,aa0regressed]];

Label[Next2]; *)

Print["Nrepairs: ", Nrepairs];
Print["Nreplacements: ", Nreplacements];

aaAppended = AppendTo[aaAppended, aa];
NcylesAppended = AppendTo[NcylesAppended, Ncycles];
Print["Ncycles Between Initial Growth and Impact: ", Ncycles];
Print["NcyclesatImpact: ", Impact1 * Nincrement];

Do[

 BetaCompounded = 1.12 - 0.231 * (aa / ChordatImpact) + 10.550 * (aa / ChordatImpact)^2 -
   21.710 * (aa / ChordatImpact)^3 + 30.382 * (aa / ChordatImpact)^4;
 BetaCompoundedAppend = AppendTo[BetaCompoundedAppend, BetaCompounded];
 DeltaK = BetaCompounded * DeltaSigma * (Pi * aa)^0.5;

 σResidual = 0;
 If[ImpactSide == "Pressure Side",
  If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 1, σResidual =
     -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) + 117 116.263407782 * aa + (aa -
              0.000335) * (aa - 0.000335) * -395 474 551.758495 + (aa - 0.000335) * (aa - 0.000335) *
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
        (aa - 0.000335) * 600 608 204 212.839 + (aa - 0.000335) * (aa - 0.000335) *
        (aa - 0.000335) * (aa - 0.000335) * -313 392 042 189 632],
      Abs[((-91.388930233299) + 163 881.618156941 * aa + (aa - 0.000335) *
          (aa - 0.000335) * -621 710 641.059832 + (aa - 0.000335) * (aa - 0.000335) *
          (aa - 0.000335) * 784 096 231 132.582 + (aa - 0.000335) * (aa - 0.000335) *
          (aa - 0.000335) * (aa - 0.000335) * -181 928 735 061 177) - ((-51.0322827764947) +
         73 621.0917592117 * aa + (aa - 0.000335) * (aa - 0.000335) * -186 084 270.781473 +
          (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * 369 136 849 298.813 +
          (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
           -331 072 585 461 178)] / 6], RandomReal[]] * If[LSPTreatment == 1, 0.75,
    If[LSPTreatment == 2, 1, If[LSPTreatment == 3, 3, If[LSPTreatment == 0, 0]]]]],
  If[ImpactSide == "Suction Side", If[0.29 ≤ Rimpact ≤ 0.39, If[0 ≤ XpercentChord ≤ 0.80,
   σResidual = -InverseCDF[NormalDistribution[Abs[(-70.7331771213364) + 117 116.263407782 *
          aa + (aa - 0.000335) * (aa - 0.000335) * -395 474 551.758495 + (aa - 0.000335) *
          (aa - 0.000335) * (aa - 0.000335) * 600 608 204 212.839 + (aa - 0.000335) *
          (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -313 392 042 189 632],
       Abs[((-91.388930233299) + 163 881.618156941 * aa + (aa - 0.000335) * (aa - 0.000335) *
           -621 710 641.059832 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) *
          784 096 231 132.582 + (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * (aa -
            0.000335) * -181 928 735 061 177) - ((-51.0322827764947) + 73 621.0917592117 *
          aa + (aa - 0.000335) * (aa - 0.000335) * -186 084 270.781473 + (aa - 0.000335) *
          (aa - 0.000335) * (aa - 0.000335) * 369 136 849 298.813 + (aa - 0.000335) *
          (aa - 0.000335) * (aa - 0.000335) * (aa - 0.000335) * -331 072 585 461 178)] / 6],
       RandomReal[]] * If[LSPTreatment == 1, 0.75, If[LSPTreatment == 2, 1,
       If[LSPTreatment == 3, 3, If[LSPTreatment == 0, 0]]]]]];

Kmax = BetaCompounded * If[aa ≤ 0.00091, AxialσMax + σResidual * 10^6, AxialσMax] * (Pi * aa)^0.5;
dadN = (0.0000000000052) * ((DeltaK / 10^6)^2.5) * ((Kmax / 10^6)^0.67);
DeltaNcycles = Nincremont;
daa = dadN * DeltaNcycles;
aa = aa + daa;
If[aa === ComplexInfinity || Head[aa] === Complex, aa = Re[aa]];
Ncycles = 11 * Nincremont;
LevelIIIDamageRandomSeed = RandomReal[];
If[EngineIntervalInspections == 1,
 If[Ncycles == NcyclesIntervalInspection && aaD ≤ aa < aac, Nrepairs = Nrepairs + 1;
  aa = EIFSa, If[LevelIIIDamageRandomSeed ≥ 4 * 10^-5,
   If[Ncycles == NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
    aa = EIFSa], If[LevelIIIDamageRandomSeed < 4 * 10^-5 && Ncycles ==
      NcyclesIntervalInspection && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]]];
 If[LevelIIIDamageRandomSeed > 4 * 10^-5,
  If[Ncycles < NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
   aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles <
      NcyclesIntervalInspection && aa ≥ aac, NEngineReplacements = NEngineReplacements + 1;
    aa = EIFSa;
    Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], 11]]];
 If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[NcyclesIntervalInspection * 1.5 > Ncycles >
     NcyclesIntervalInspection && aa ≥ aac, Nreplacements = Nreplacements + 1;
   aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5,
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
22 |
          If[NcyclesIntervalInspection * 1.5 > Ncycles > NcyclesIntervalInspection && aa ≥ aac,
           NEngineroplacements = NEngineroplacements + 1;
           aa = EIFSa;
           Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]]];
         If[Ncycles == NcyclesIntervalInspection * 1.5 && aaD ≤ aa < aac, Nrepairs = Nrepairs + 1;
          aa = EIFSa, If[LevelIIIDamageRandomSeed ≥ 4 * 10^-5,
           If[Ncycles == NcyclesIntervalInspection * 1.5 && aa ≥ aac, Nreplacements = Nreplacements + 1;
            aa = EIFSa], If[LevelIIIDamageRandomSeed < 4 * 10^-5 &&
             Ncycles == NcyclesIntervalInspection * 1.5 && aa ≥ aac,
            NEngineroplacements = NEngineroplacements + 1;
            aa = EIFSa;
            Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]]];
         If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[Ntakeoffs * 2 * 60 * 60 * StressFrequency > Ncycles >
             NcyclesIntervalInspection * 1.5 && aa ≥ aac, Nreplacements = Nreplacements + 1;
           aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5,
           If[Ncycles > NcyclesIntervalInspection * 1.5 && aa ≥ aac,
            NEngineroplacements = NEngineroplacements + 1;
            aa = EIFSa;
            Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]],
         If[EngineIntervalInspections == 0, If[LevelIIIDamageRandomSeed > 4 * 10^-5, If[Ncycles <
              Ntakeoffs * 2 * 60 * 60 * StressFrequency && aa ≥ aac, Nreplacements = Nreplacements + 1;
            aa = EIFSa], If[LevelIIIDamageRandomSeed ≤ 4 * 10^-5, If[Ncycles < Ntakeoffs * 2 *
                60 * 60 * StressFrequency && aa ≥ aac, NEngineroplacements = NEngineroplacements + 1;
             aa = EIFSa;
             Print[Style["ENGINE WAS REPLACED ON TAKEOFF: ", 36, Red], ii]]]]];
        If[aac ≤ aa || Ntakeoffs * 2 * 60 * 60 * StressFrequency == Ncycles,
         Print["ith takeoff during Natural Growth when a >= aac or
            Ncycles = Ntakeoffs*2*60*60*StressFrequency: ", ii];
         i = i + 1;
         NcyclesatbladeroplacementTotal = AppendTo[NcyclesatbladeroplacementTotal, Ncycles];
         Nbladeroplacements = Nbladeroplacements + 1;
         Ncycles = 0,
         aaAppended = AppendTo[aaAppended, aa];
         NcylesAppended = AppendTo[NcylesAppended, Ncycles]],
        {ii, Impacti + 1, Ntakeoffs}];

       If[Nrepairs > 0, CostIncreaseDuetoEngineEfficiencyDrogradation =
         (0.005 + RandomReal[] * 0.005) * 2 * CostofFuelperFlight * (Ntakeoffs - Impacti)];
       Print["CostIncreaseDuetoEngineEfficiencyDrogradation: ",
        CostIncreaseDuetoEngineEfficiencyDrogradation];


       Print["Nrepairs: ", Nrepairs];
       Print["Nreplacements: ", Nreplacements];
       Print["NEngineroplacements: ", NEngineroplacements]


        Goto[End1]]], Goto[Next4]], Goto[Next4]]; Label[Next4], {i, 1, Ntakeoffs}]];

Label[End1];
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
LifeTimeRepairs = NRepairs - LifeTimeRepairs;
Print["LifeTimeRepairs: ", LifeTimeRepairs];
LifeTimeReplacements = NReplacements - LifeTimeReplacements;
Print["LifeTimeReplacements: ", LifeTimeReplacements];
LifeTimeEngineReplacements = NEngineReplacements - LifeTimeEngineReplacements;
Print["LifeTimeEngineReplacements: ", LifeTimeEngineReplacements];

LifeTimeFuelCostDueToEfficiencyDegradation = CostIncreaseDuetoEngineEfficiencyDegradation;

NRepairsAppended = AppendTo[NRepairsAppended, LifeTimeRepairs];
NReplacementsAppended = AppendTo[NReplacementsAppended, LifeTimeReplacements];
NEngineReplacementsAppended =
 AppendTo[NEngineReplacementsAppended, LifeTimeEngineReplacements];

(* Life Time Costing Models *)

SCHCALRepairs = LifeTimeRepairs * (FLTS * NA * QTY * AVEFLTHR * RECMHRSRepairs * BurdenedDIRLAB) /
    (NCyclesIntervalInspection / (60 * 60 * StressFrequency));
SCHCALRepairsAppended = AppendTo[SCHCALRepairsAppended, SCHCALRepairs];

SCHEDCAMATRepairs = LifeTimeRepairs * (FLTS * NA * QTY * AVEFLTHR * SCHCAMATRepairs) /
    (NCyclesIntervalInspection / (60 * 60 * StressFrequency));
SCHEDCAMATRepairsAppended = AppendTo[SCHEDCAMATRepairsAppended, SCHEDCAMATRepairs];

LifeTimeRepairsCosts = SCHCALRepairs + SCHEDCAMATRepairs;
LifeTimeRepairsCostsAppended = AppendTo[LifeTimeRepairsCostsAppended, LifeTimeRepairsCosts];

(* Component Replacements and Associated Labor *)
LRReplacements =
 LifeTimeReplacements * (FLTS * NA * QTY * AVEFLTHR * RECMHRSReplacements * BurdenedDIRLAB) /
    (NCyclesIntervalInspection / (60 * 60 * StressFrequency)); (* Component Replacements *)
LRReplacementsAppended = AppendTo[LRReplacementsAppended, LRReplacements];

MATReplacements = LifeTimeReplacements * (FLTS * NA * QTY * AVEFLTHR * CAMATReplacements) /
    (NCyclesIntervalInspection / (60 * 60 * StressFrequency));
CAMATReplacementsAppended = AppendTo[CAMATReplacementsAppended, MATReplacements];

LifeTimeReplacementsCosts = LRReplacements + MATReplacements;
LifeTimeReplacementsCostsAppended =
 AppendTo[LifeTimeReplacementsCostsAppended, LifeTimeReplacementsCosts];

(* Engine Replacements *)
LEngineReplacements =
 LifeTimeEngineReplacements * (FLTS * NA * QTY * AVEFLTHR * MHRSEngineReplacements * BurdenedDIRLAB) /
    (NCyclesIntervalInspection / (60 * 60 * StressFrequency));
LEngineReplacementsAppended = AppendTo[LEngineReplacementsAppended, LEngineReplacements];

MATEngineReplacements =
 LifeTimeEngineReplacements * (FLTS * NA * QTY * AVEFLTHR * CAMATEngineReplacements) /
    (NCyclesIntervalInspection / (60 * 60 * StressFrequency));
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
24 |
MATEngineReplacementsAppended = AppendTo[MATEngineReplacementsAppended, MATEngineReplacements];

TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisitionAppended =
 AppendTo[TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisitionAppended,
  TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisition];

Print["LifeTimeFuelCostDueToEfficiencyDegradation: ",
 LifeTimeFuelCostDueToEfficiencyDegradation];

Print["LifeTimeCostofScheduledInspections: ", SCHINCOST];
Print["LifeTimeCostofScheduledRepairLabor: ", SCHCALrepairs];
Print["LifeTimeCostofUnscheduledReplacementLabor: ", LreplacementsT];
Print["LifeTimeCostofScheduledRepairMaterial: ", SCHEDCAMATrepairs];
Print["LifeTimeCostofUnscheduledRepairMaterial: ", MATreplacements];
Print["TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisition: ",
 TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisition];
Print["LifeTimeCostofEngineReplacementLabor: ", LEngineReplacements];
Print["LifeTimeCostofEngineReplacement: ", MATEngineReplacements];
Print["LifeTimeFuelCostDueToEfficiencyDegradation: ",
 LifeTimeFuelCostDueToEfficiencyDegradation];

TotalLifeTimeCosts = SCHINCOST + SCHCALrepairs + LreplacementsT + SCHEDCAMATrepairs +
  MATreplacements + TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisition +
  LEngineReplacements + MATEngineReplacements + NA * LifeTimeFuelCostDueToEfficiencyDegradation;
Print["TotalLifeTimeCosts: ", TotalLifeTimeCosts];
TotalLifeTimeCostsAppended = AppendTo[TotalLifeTimeCostsAppended, TotalLifeTimeCosts];


NbladeReplacementsAppended = AppendTo[NbladeReplacementsAppended, NbladeReplacements];
aavsNcycles = Transpose[{NcyclesAppended, aaAppended}];
Print[
 "DOE Settings for Number of Runway Visual Inspections, Number of Runway Debris Sweeps, LSP
    Condition and Determination if Engine Interval Inspections
    Take Place: ", CrackGrowthModel[[i1]]];
Print[ListPlot[aavsNcycles, GridLines → Automatic, PlotLegends → "Crack Growth a Including
     Post Impact", AxesLabel → ("Stress Cycles (N)", "CrackDepth a (meters)")]];
aavsNcyclesNaturalGrowth = Transpose[{NcyclesNaturalGrowthAppended, aaAppendedNaturalGrowth}];
Print[ListPlot[aavsNcyclesNaturalGrowth, GridLines → Automatic, PlotLegends →
   "Natural Crack Growth a", AxesLabel → ("Stress Cycles (N)", "CrackDepth a (meters)")]];
Print[ListPlot[{aavsNcycles, aavsNcyclesNaturalGrowth}, GridLines → Automatic,
  PlotLegends → {"Crack Growth a Including Post Impact", "Natural Crack Growth a"},
  AxesLabel → ("Stress Cycles (N)", "CrackDepth a (meters)")]];

(*Print["NbladeReplacements: ",NbladeReplacements];
Print["NcyclesatbladeReplacement: ",NcyclesatbladeReplacement];*)
(*Print["aavsNcycles: ",aavsNcycles];
Print["NcyclesAppendedTotal: ",NcyclesAppendedTotal];*)
ClearAll[Ncycles, NcyclesNaturalGrowth];
Label[End2],
(n, 1, 101)];
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
(*Print["aa0regressedAppended: ",aa0regressedAppended]
  aa0regressed90thPercentile=Quantile[aa0regressedAppended,0.90];
Print["aa0regressed90thPercentile: ",aa0regressed90thPercentile]
  Print["NbladereplacementsAppended: ",NbladereplacementsAppended]
  Print["NcyclesatbladereplacemenT: ",NcyclesatbladereplacemenT]
  Histogram[NbladereplacementsAppended,AxesLabel→{"NbladereplacementsAppended","Frequency"}]
  Histogram[NcyclesatbladereplacemenT,AxesLabel→{"NcyclesatbladereplacemenT","Frequency"}]
  Ncycles90thPercentileInspection=Quantile[NcyclesatbladereplacemenT,0.90]/2;
Print["Ncycles90thPercentileInspection: ",Ncycles90thPercentileInspection]*)
Print[
 "DOE Settings for Number of Runway Visual Inspections, Number of Runway Debris Sweeps, LSP
   Condition and Determination if Engine Interval Inspections
   Take Place: ", CrackGrowthModel[[i1]]];
Print["Nimpacts: ", Nimpacts];
Print["Nrepairs: ", Nrepairs];
Print["Nreplacements: ", Nreplacements];
Print["NEnginereplacements: ", NEnginereplacements];
Print["NrepairsAppended: ", NrepairsAppended];
Print["NreplacementsAppended: ", NreplacementsAppended];
Print["NEnginereplacementsAppended: ", NEnginereplacementsAppended];
Print["SCHCALrepairsAppended: ", SCHCALrepairsAppended];
Print["LreplacementsAppended: ", LreplacementsAppended];
Print["SCHEDCAMATrepairsAppended: ", SCHEDCAMATrepairsAppended];
Print["CAMATreplacementsAppended: ", CAMATreplacementsAppended];
Print["LEnginereplacementsAppended: ", LEnginereplacementsAppended];
Print["MATEnginereplacementsAppended: ", MATEnginereplacementsAppended];
Print["TotalLifeTimeCostsAppended: ", TotalLifeTimeCostsAppended];
Histogram[TotalLifeTimeCostsAppended,
 AxesLabel → {"TotalLifeTimeCostsAppended Using Visual Inspections", "Frequency"}];
NrepairsAppendedDOE = AppendTo[NrepairsAppendedDOE, NrepairsAppended];
NreplacementsAppendedDOE = AppendTo[NreplacementsAppendedDOE, NreplacementsAppended];
NEnginereplacementsAppendedDOE =
 AppendTo[NEnginereplacementsAppendedDOE, NEnginereplacementsAppended];
NimpactsAppendedDOE = AppendTo[NimpactsAppendedDOE, Nimpacts];
TotalLifeTimeCostsAppendedDOE =
 AppendTo[TotalLifeTimeCostsAppendedDOE, TotalLifeTimeCostsAppended];
TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisitionAppendedDOE =
 AppendTo[TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisitionAppendedDOE,
  TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisitionAppended];
CAMATreplacementsAppendedDOE = AppendTo[CAMATreplacementsAppendedDOE, CAMATreplacementsAppended];
LifeTimereplacementscostsAppendedDOE =
 AppendTo[LifeTimereplacementscostsAppendedDOE, LifeTimereplacementscostsAppended];
LifeTimerepairscostsAppendedDOE =
 AppendTo[LifeTimerepairscostsAppendedDOE, LifeTimerepairscostsAppended],
{i1, 1, InputFileLength}]

Print["DOE Settings for Number of Runway Visual Inspections, Number of Runway Debris Sweeps, LSP
   Condition and Determination if Engine Interval Inspections Take Place: ", CrackGrowthModel]
Print["TotalLifeTimeCostsAppendedDOE: ", TotalLifeTimeCostsAppendedDOE]
Print["NrepairsAppendedDOE: ", NrepairsAppendedDOE]
Print["NreplacementsAppendedDOE: ", NreplacementsAppendedDOE]
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
26 |
Print["NEngineReplacementsAppendedDOE: ", NEngineReplacementsAppendedDOE]
Print["NImpactsAppendedDOE: ", NImpactsAppendedDOE]
Print["TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisitionAppendedDOE: ",
 TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisitionAppendedDOE]
Print["LifeTimeReplacementscostsAppendedDOE: ", LifeTimeReplacementscostsAppendedDOE]
Print["LifeTimeRepairscostsAppendedDOE: ", LifeTimeRepairscostsAppendedDOE]

Show[Histogram[Table[NRepairsAppendedDOE[[1]], {1, 1, Length[NRepairsAppendedDOE]}],
  PlotRange → All, PlotLabel → "NRepairsAppended", ChartStyle → "Pastel",
  AxesLabel → {"NRepairsAppended", "Frequency"}]]

Show[Histogram[Table[NReplacementsAppendedDOE[[1]], {1, 1, Length[NReplacementsAppendedDOE]}],
  PlotRange → All, PlotLabel → "NReplacementsAppended", ChartStyle → "Pastel",
  AxesLabel → {"NReplacementsAppended", "Frequency"}]]

Show[Histogram[
  Table[NEngineReplacementsAppendedDOE[[1]], {1, 1, Length[NEngineReplacementsAppendedDOE]}],
  PlotRange → All, PlotLabel → "NEngineReplacementsAppended", ChartStyle → "Pastel",
  AxesLabel → {"NEngineReplacementsAppended", "Frequency"}]]

Show[Histogram[Table[NImpactsAppendedDOE[[1]], {1, 1, Length[NImpactsAppendedDOE]}],
  PlotRange → All, PlotLabel → "NImpactsAppended", ChartStyle → "Pastel",
  AxesLabel → {"NImpactsAppended", "Frequency"}]]

Show[
 Histogram[Table[TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisitionAppendedDOE[[1]],
   {1, 1, Length[TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisitionAppendedDOE]}],
  PlotRange → All, PlotLabel →
   "TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisitionAppended",
  ChartStyle → "Pastel", AxesLabel →
   {"TotalCostofVisualInspectionsandRunwaySweepandSweeperAcquisitionAppended", "Frequency"}]]

Show[
 Histogram[Table[TotalLifeTimeCostsAppendedDOE[[1]], {1, 1, Length[TotalLifeTimeCostsAppendedDOE]}],
  PlotRange → All, PlotLabel → "TotalLifeTimeCostsUsingVisualInspections", ChartStyle → "Pastel",
  AxesLabel → {"TotalLifeTimeCosts", "Frequency"}, ChartLegends → Automatic]]

Table[Show[Histogram[TotalLifeTimeCostsAppendedDOE[[1]], PlotRange → All,
   PlotLabel → "TotalLifeTimeCostsUsingVisualInspections", ChartStyle → "Pastel",
   AxesLabel → {"TotalLifeTimeCosts", "Frequency"}, ChartLegends → "DOE Setting" -> i]],
 {i, 1, Length[TotalLifeTimeCostsAppendedDOE]}]



Do[TotalLifeTimeCostsAppendedDOEVisualMean =
   AppendTo[TotalLifeTimeCostsAppendedDOEVisualMean, Mean[TotalLifeTimeCostsAppendedDOE[[i]]]];
  TotalLifeTimeCostsAppendedDOEVisualStd = AppendTo[TotalLifeTimeCostsAppendedDOEVisualStd,
    StandardDeviation[TotalLifeTimeCostsAppendedDOE[[i]]]];
  TotalLifeTimeCostsAppendedDOEVisualMeanvsDOESetting =
   AppendTo[TotalLifeTimeCostsAppendedDOEVisualMeanvsDOESetting,
    {i, Mean[TotalLifeTimeCostsAppendedDOE[[i]]]}];
```


Crack Growth and LCC Code – Visual Inspections, 1.33 mm Steel FOD (cont.)

```
  TotalLifeTimeCostsAppendedDOEVisualStdvsDOESetting =
   AppendTo[TotalLifeTimeCostsAppendedDOEVisualStdvsDOESetting,
    {i, StandardDeviation[TotalLifeTimeCostsAppendedDOE[[i]]]}],
  {i, 1, Length[TotalLifeTimeCostsAppendedDOE]}];

Do[NreplacementsvsDOESettingAppendedVisualSum =
   AppendTo[NreplacementsvsDOESettingAppendedVisualSum,
    {ii, Sum[NreplacementsAppendedDOE[[ii]][[i]], {i, 1, Length[NreplacementsAppendedDOE[[ii]]]}]}],
  {ii, 1, Length[NreplacementsAppendedDOE]}];

Do[NrepairsvsDOESettingAppendedVisualSum = AppendTo[NrepairsvsDOESettingAppendedVisualSum,
    {ii, Sum[NrepairsAppendedDOE[[ii]][[i]], {i, 1, Length[NrepairsAppendedDOE[[ii]]]}]}],
  {ii, 1, Length[NrepairsAppendedDOE]}];


ListPlot[TotalLifeTimeCostsAppendedDOEVisualMeanvsDOESetting,
 AxesLabel → {"DOE Setting", "MeanTotalLifeTimeCosts"},
 PlotLegends → "Visual Inspections 1.33 mm Impactor 100 Replicates"]
ListPlot[TotalLifeTimeCostsAppendedDOEVisualStdvsDOESetting,
 AxesLabel → {"DOE Setting", "StdTotalLifeTimeCosts"},
 PlotLegends → "Visual Inspections 1.33 mm Impactor 100 Replicates"]
ListPlot[NreplacementsvsDOESettingAppendedVisualSum,
 AxesLabel → {"DOE Setting", "Sum of Component Replacements"},
 PlotLegends → "Visual Inspections 1.33 mm Impactor 100 Replicates"]
ListPlot[NrepairsvsDOESettingAppendedVisualSum,
 AxesLabel → {"DOE Setting", "Sum of Component Repairs"},
 PlotLegends → "Visual Inspections 1.33 mm Impactor 100 Replicates"]

TotalLifeTimeCostsStandardDeviationofMeans =
  StandardDeviation[TotalLifeTimeCostsAppendedDOEVisualMean];
StdofMeansandNumberofReplicatesList = {100, TotalLifeTimeCostsStandardDeviationofMeans};

Print["TotalLifeTimeCostsStandardDeviationofMeans: ", TotalLifeTimeCostsStandardDeviationofMeans]
Print["StdofMeansandNumberofReplicatesList: ", StdofMeansandNumberofReplicatesList]
```

APPENDIX G: PARTICLE IMPACT FITS & DISTRIBUTIONS

Recall that for the fits derived from data generated with the In-Engine Particle Kinetics model four input factors control the experimentation (Section 4.6): engine/ground clearance, stagnation point ground distance to fan face, inlet Mach number and Head Wind – the external geometry of the engine and the vortex ground conditions under the experimental conditions for the case study lead to strong vortices that will aspirate debris if it is located inside the ground vortex stagnation point (Sections 4.5 and 4.6). The experiment is run for each condition (DOE setting) in the In-Engine Particle Kinetics model simulation (Appendix E) – this simulation code contains the inlet velocity and location of the ingested vortex at fan face fits from the Aspiration Model simulation run for the same experimental factors and levels as those of the case study, Beta distributions for flow aero/thermos dynamics parameters from 10,000 runs (at 10,000 random inlet Mach numbers) of the Meanline Analysis model. The experimentation input factors are varied at three levels for a Full Factorial of 81 settings (see Table 5.1). To expand the coverage of the experiments and to consider internal experimental points 1,000 (points are evenly spaced) additional settings were concatenated to the Full Factorial 81 settings – these additional settings are called Space Filling Latin Hypercube designs, where levels are spaced evenly from lower bound to upper bound of each factor. The experiments generate data that is used to develop fits for axial stress, chord length, radial location and local thickness at impact site and Beta distributions for Non-dim chord location and depth of penetration at impact site (recall that impact velocity and depth of penetration are indexed statistically) - the bounds of the inputs over which the fits and distributions have been validated are presented again in Table G.1. Recall that because rotor blades are not

stationary their random location at the moment of impact relative to a FOD particle presents Beta distributions rather than fits for Non-dim chord location and depth of penetration at impact. The Beta distributions are seeded with a random number (from uniform distribution with range of 0 to 1) indexed to the inlet Mach number presented Table G.1. This appendix presents the fits and Beta distributions along with associated statistical measures.

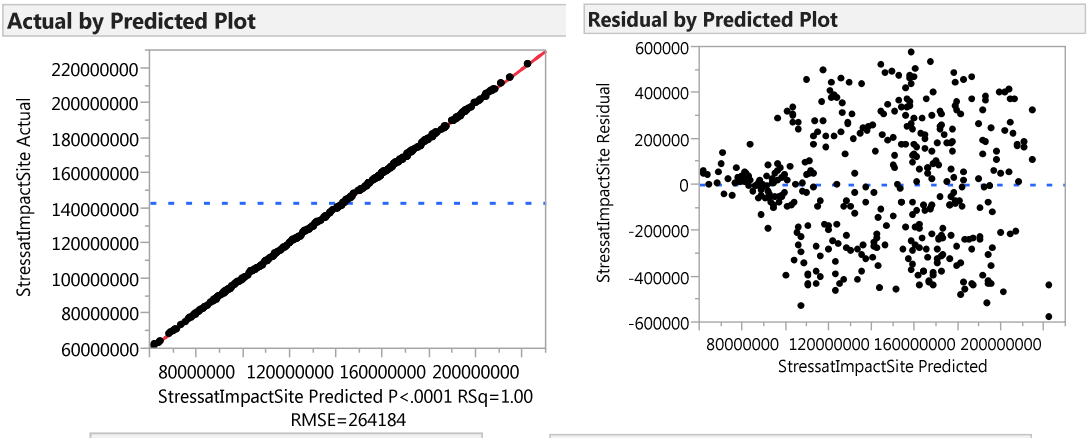
Table G.1: Particle Impact Input Bounds

Ground Clearance, h (m)		
0.24	0.285	0.33
Ground Distance Clearance to Fan Face (m)		
0.5	0.75	1
Inlet Mach Number (M_i)		
0.45	0.475	0.5
Wind Velocity (Head Wind, U_∞) (m/s)		
9.357	10.6785	12

Axial stress at impact fit and goodness of fit for both particle sizes:

$$\sigma_{Axial,fit,1.33mm} = (1063168750.67919 + 515059837.719837 * \mathrm{h} + -366022.249681334 * \mathrm{GroundDistance} + -2123990818.12455 * \mathrm{InletMachNumber} + -5893409.84568672 * \mathrm{HeadWind} + (\mathrm{h} - 0.285513648014925) * ((\mathrm{h} - 0.285513648014925) * -115914736.246555) + (\mathrm{h} - 0.285513648014925) * ((\mathrm{GroundDistance} - 0.727088406885572) * -3860239.90929367) + (\mathrm{GroundDistance} - 0.727088406885572) * ((\mathrm{GroundDistance} - 0.727088406885572) * 601596.125270522) + (\mathrm{h} - 0.285513648014925) * ((\mathrm{InletMachNumber} - 0.472574813666666) * -1770961724.2888) + (\mathrm{GroundDistance} - 0.727088406885572) * ((\mathrm{InletMachNumber} - 0.472574813666666) * -3601984.15779136) + (\mathrm{InletMachNumber} - 0.472574813666666) * ((\mathrm{InletMachNumber} -$$

$$0.472574813666666) * -1852995080.65505) + (h - 0.285513648014925) * ((HeadWind - 10.6542844452363) * 1815221.56210402) + (GroundDistance - 0.727088406885572) * ((HeadWind - 10.6542844452363) * -25325.6256442044) + (InletMachNumber - 0.472574813666666) * ((HeadWind - 10.6542844452363) * -12296059.8523198) + (HeadWind - 10.6542844452363) * ((HeadWind - 10.6542844452363) * 222250.495346287)) * (10500/14000)^2 \quad (G.1)$$



Summary of Fit

RSquare	0.999954
RSquare Adj	0.999952
Root Mean Square Error	264183.9
Mean of Response	1.431e+8
Observations (or Sum Wgts)	402

Analysis of Variance

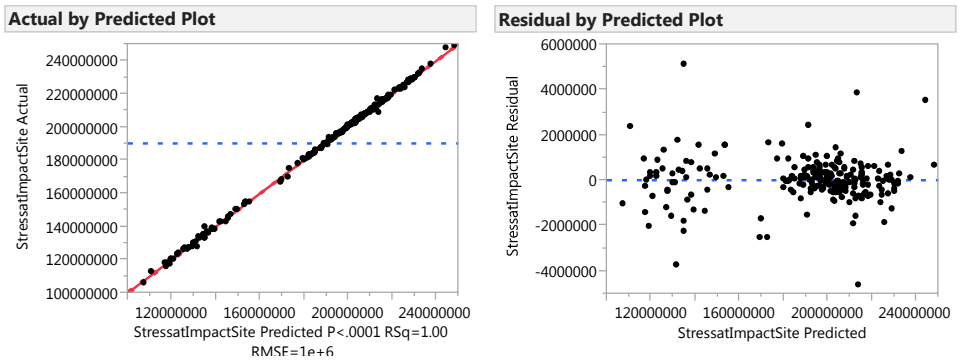
Source	DF	Sum of Squares	Mean Square	F Ratio
Model	14	5.8455e+17	4.175e+16	598249.8
Error	387	2.701e+13	6.979e+10	Prob > F
C. Total	401	5.8458e+17		<.0001*

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
InletMachNumber	-2.124e+9	868102.9	-2447	<.0001*
h	515059838	510554.5	1008.8	<.0001*
HeadWind	-5893410	17031.35	-346.0	<.0001*
(h-0.28551)*(InletMachNumber-0.47257)	-1.771e+9	31426213	-56.35	<.0001*
(InletMachNumber-0.47257)*(InletMachNumber-0.47257	-1.853e+9	66491052	-27.87	<.0001*
(InletMachNumber-0.47257)*(HeadWind-10.6543)	-12296060	1032998	-11.90	<.0001*
(HeadWind-10.6543)*(HeadWind-10.6543)	222250.5	24104.29	9.22	<.0001*
(h-0.28551)*(h-0.28551)	-1.159e+8	20874045	-5.55	<.0001*
GroundDistance	-366022.2	93080.01	-3.93	<.0001*
(h-0.28551)*(HeadWind-10.6543)	1815221.6	651471	2.79	0.0056*
(h-0.28551)*(GroundDistance-0.72709)	-3860240	3405219	-1.13	0.2577
(GroundDistance-0.72709)*(GroundDistance-0.72709)	601596.13	655735.8	0.92	0.3595
(GroundDistance-0.72709)*(InletMachNumber-0.47257)	-3601984	5521588	-0.65	0.5146
(GroundDistance-0.72709)*(HeadWind-10.6543)	-25325.63	117562	-0.22	0.8296

Figure G.1: Axial Stress Goodness of Fit, 1.33 mm FOD

$$\sigma_{Axial,fit,3.2mmFOD} = (1082250836.78934 + 351147159.372234 * \text{h} + -1709991.81602705 * \text{GroundDistance} + -1992048164.33473 * \text{InletMachNumber} + -5611743.67562871 * \text{HeadWind} + (\text{h} - 0.284538338355556) * ((\text{h} - 0.284538338355556) * -709179047.112527) + (\text{h} - 0.284538338355556) * ((\text{GroundDistance} - 0.794982760546667) * -101165634.43643) + (\text{GroundDistance} - 0.794982760546667) * ((\text{GroundDistance} - 0.794982760546667) * -29528228.0823277) + (\text{h} - 0.284538338355556) * ((\text{InletMachNumber} - 0.466504282097778) * 1648575802.45814) + (\text{GroundDistance} - 0.794982760546667) * ((\text{InletMachNumber} - 0.466504282097778) * 215166090.472403) + (\text{InletMachNumber} - 0.466504282097778) * ((\text{InletMachNumber} - 0.466504282097778) * -1367013077.27309) + (\text{h} - 0.284538338355556) * ((\text{HeadWind} - 10.7592392263289) * -28424792.5131628) + (\text{GroundDistance} - 0.794982760546667) * ((\text{HeadWind} - 10.7592392263289) * 688182.698049512) + (\text{InletMachNumber} - 0.466504282097778) * ((\text{HeadWind} - 10.7592392263289) * 11326309.7653247) + (\text{HeadWind} - 10.7592392263289) * ((\text{HeadWind} - 10.7592392263289) * 544325.197231379)) * (10500/14000)\text{^}2 \quad \text{(G.2)}$$



Summary of Fit

RSquare	0.999104
RSquare Adj	0.999045
Root Mean Square Error	1018308
Mean of Response	1.901e+8
Observations (or Sum Wgts)	225

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	14	2.4288e+17	1.735e+16	16730.27
Error	210	2.1776e+14	1.037e+12	Prob > F
C. Total	224	2.431e+17		<.0001*

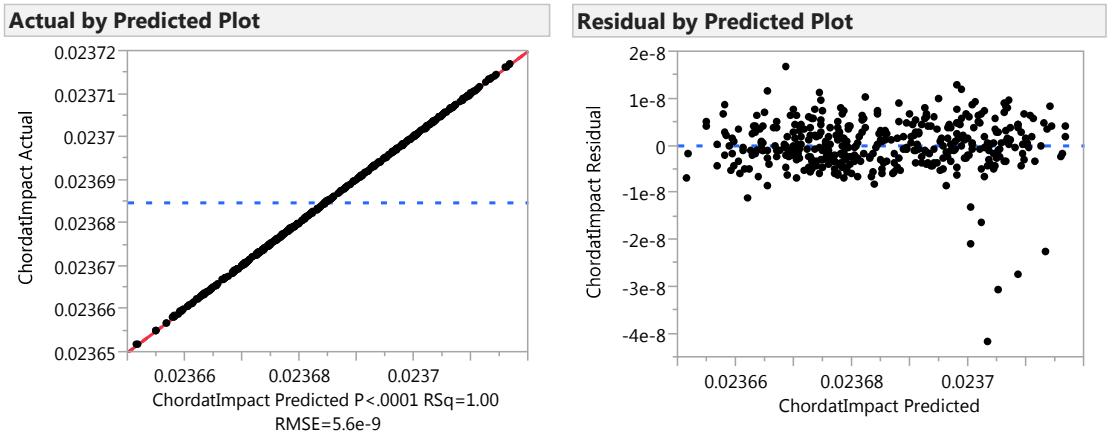
Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
InletMachNumber	-1.992e+9	8346243	-238.7	<.0001*
h	351147159	2751911	127.60	<.0001*
HeadWind	-5611744	86843.07	-64.62	<.0001*
(h-0.28454)*(InletMachNumber-0.4665)	1.6486e+9	1.682e+8	9.80	<.0001*
(h-0.28454)*(HeadWind-10.7592)	-28424793	3253269	-8.74	<.0001*
(GroundDistance-0.79498)*(GroundDistance-0.79498)	-29528228	3817542	-7.73	<.0001*
(GroundDistance-0.79498)*(InletMachNumber-0.4665)	215166090	29485951	7.30	<.0001*
(h-0.28454)*(h-0.28454)	-7.092e+8	1.126e+8	-6.30	<.0001*
(h-0.28454)*(GroundDistance-0.79498)	-1.012e+8	18522382	-5.46	<.0001*
(HeadWind-10.7592)*(HeadWind-10.7592)	544325.2	119999.5	4.54	<.0001*
(InletMachNumber-0.4665)*(InletMachNumber-0.4665	-1.367e+9	3.977e+8	-3.44	0.0007*
GroundDistance	-1709992	544159.9	-3.14	0.0019*
(InletMachNumber-0.4665)*(HeadWind-10.7592)	11326310	5475455	2.07	0.0398*
(GroundDistance-0.79498)*(HeadWind-10.7592)	688182.7	595412.5	1.16	0.2491

Figure G.2: Axial Stress Goodness of Fit, 3.2 mm FOD

Chord length at impact fit for both sizes:

$$\text{ChordatImpact}_{fit,1.33mmFOD} = 0.0233114081179261 + -0.000209752221913733 * \text{h} + 0.0000001776214368989 * \text{GroundDistance} + 0.000862890506036301 * \text{InletMachNumber} + 0.0000023869436839637 * \text{HeadWind} + (\text{h} - 0.285513648014925) * ((\text{h} - 0.285513648014925) * 0.000002167677195839) + (\text{h} - 0.285513648014925) * ((\text{GroundDistance} - 0.727088406885572) * 0.0000002712313406962) + (\text{GroundDistance} - 0.727088406885572) * ((\text{GroundDistance} - 0.727088406885572) * -0.0000004849014682213) + (\text{h} - 0.285513648014925) * ((\text{InletMachNumber} - 0.472574813666666) * 0.00106043173745756) + (\text{GroundDistance} - 0.727088406885572) * ((\text{InletMachNumber} - 0.472574813666666) * 0.0000017569184995802) + (\text{InletMachNumber} - 0.472574813666666) * ((\text{InletMachNumber} - 0.472574813666666) * -0.0000097442681092735) + (\text{h} - 0.285513648014925) * ((\text{HeadWind} - 10.6542844452363) * -0.0000000417742351687) + (\text{GroundDistance} - 0.727088406885572) * ((\text{HeadWind} - 10.6542844452363) * -0.0000000044377544096) + (\text{InletMachNumber} - 0.472574813666666) * ((\text{HeadWind} - 10.6542844452363) * -0.0000000678667866207) + (\text{HeadWind} - 10.6542844452363) * ((\text{HeadWind} - 10.6542844452363) * -0.0000000800667122655) \quad \text{(G.3)}$$



Summary of Fit

RSquare	1
RSquare Adj	1
Root Mean Square Error	5.578e-9
Mean of Response	0.023685
Observations (or Sum Wgts)	402

Analysis of Variance

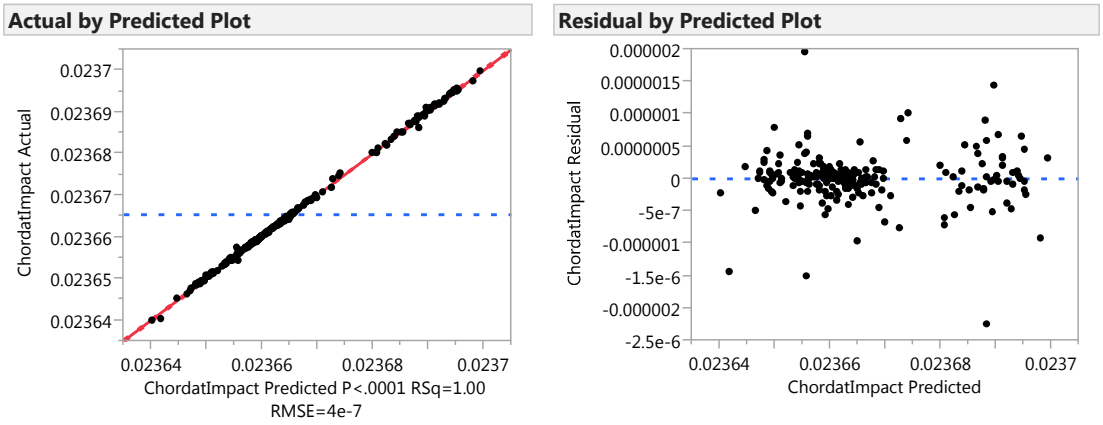
Source	DF	Sum of Squares	Mean Square	F Ratio
Model	14	9.60362e-8	6.8597e-9	2.205e+8
Error	387	1.2041e-14	3.111e-17	Prob > F
C. Total	401	9.60362e-8		<.0001*

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
InletMachNumber	0.0008629	1.833e-8	47077	<.0001*
h	-0.00021	1.078e-8	-19458	<.0001*
HeadWind	2.3869e-6	3.6e-10	6637.7	<.0001*
(h-0.28551)*(InletMachNumber-0.47257)	0.0010604	6.635e-7	1598.1	<.0001*
(HeadWind-10.6543)*(HeadWind-10.6543)	-8.007e-8	5.09e-10	-157.3	<.0001*
GroundDistance	1.7762e-7	1.965e-9	90.38	<.0001*
(GroundDistance-0.72709)*(GroundDistance-0.72709)	-4.849e-7	1.385e-8	-35.02	<.0001*
(GroundDistance-0.72709)*(InletMachNumber-0.47257)	1.7569e-6	1.166e-7	15.07	<.0001*
(InletMachNumber-0.47257)*(InletMachNumber-0.47257	-9.744e-6	1.404e-6	-6.94	<.0001*
(h-0.28551)*(h-0.28551)	2.1677e-6	4.407e-7	4.92	<.0001*
(h-0.28551)*(GroundDistance-0.72709)	2.7123e-7	7.19e-8	3.77	0.0002*
(InletMachNumber-0.47257)*(HeadWind-10.6543)	-6.787e-8	2.181e-8	-3.11	0.0020*
(h-0.28551)*(HeadWind-10.6543)	-4.177e-8	1.376e-8	-3.04	0.0026*
(GroundDistance-0.72709)*(HeadWind-10.6543)	-4.438e-9	2.482e-9	-1.79	0.0746

Figure G.3: Chord Length Goodness of Fit, 1.33 mm FOD

$$\begin{aligned}
&\text{ChordatImpact, fit, 3.2mmFOD} = 0.0232878713039394 + \\
&-0.000148933321703101 * \text{h} + 0.0000007552066696621 * \text{GroundDistance} + \\
&0.000843421121981522 * \text{InletMachNumber} + 0.000002375267088671 * \\
&\text{HeadWind} + (\text{h} - 0.284538338355556) * ((\text{h} - 0.284538338355556) * \\
&0.000278227420847353) + (\text{h} - 0.284538338355556) * ((\text{GroundDistance} - \\
&0.794982760546667) * 0.000040609498549019) + (\text{GroundDistance} - \\
&0.794982760546667) * ((\text{GroundDistance} - 0.794982760546667) * \\
&0.0000118455823244783) + (\text{h} - 0.284538338355556) * ((\text{InletMachNumber} - \\
&0.466504282097778) * -0.000407077181481378) + (\text{GroundDistance} - \\
&0.794982760546667) * ((\text{InletMachNumber} - 0.466504282097778) * \\
&-0.0000867463608088083) + (\text{InletMachNumber} - 0.466504282097778) * \\
&((\text{InletMachNumber} - 0.466504282097778) * -0.000215622649337739) + (\text{h} - \\
&0.284538338355556) * ((\text{HeadWind} - 10.7592392263289) * \\
&0.0000134983116356392) + (\text{GroundDistance} - 0.794982760546667) * \\
&((\text{HeadWind} - 10.7592392263289) * -0.0000002403651469171) + \\
&(\text{InletMachNumber} - 0.466504282097778) * ((\text{HeadWind} - 10.7592392263289) * \\
&-0.000009219181781341) + (\text{HeadWind} - 10.7592392263289) * ((\text{HeadWind} - \\
&10.7592392263289) * -0.0000002206730965183)
\end{aligned} \tag{G.4}$$



Summary of Fit

RSquare	0.999224
RSquare Adj	0.999172
Root Mean Square Error	3.959e-7
Mean of Response	0.023665
Observations (or Sum Wgts)	225

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	14	4.23582e-8	3.0256e-9	19307.40
Error	210	3.2908e-11	1.567e-13	Prob > F
C. Total	224	4.23911e-8		<.0001*

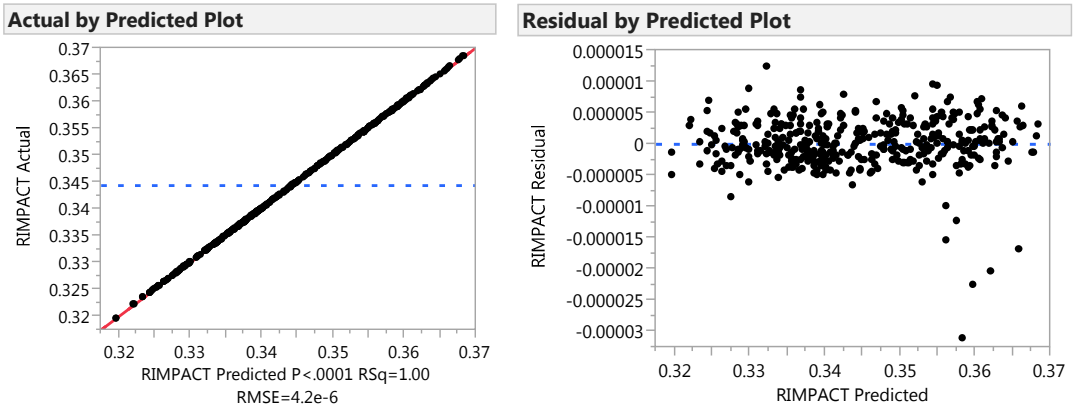
Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
InletMachNumber	0.0008434	3.245e-6	259.95	<.0001*
h	-0.000149	1.07e-6	-139.2	<.0001*
HeadWind	2.3753e-6	3.376e-8	70.36	<.0001*
(h-0.28454)*(HeadWind-10.7592)	0.0000135	1.265e-6	10.67	<.0001*
(GroundDistance-0.79498)*(GroundDistance-0.79498)	1.1846e-5	1.484e-6	7.98	<.0001*
(GroundDistance-0.79498)*(InletMachNumber-0.4665)	-8.675e-5	1.146e-5	-7.57	<.0001*
(h-0.28454)*(h-0.28454)	0.0002782	4.376e-5	6.36	<.0001*
(h-0.28454)*(InletMachNumber-0.4665)	-0.000407	6.537e-5	-6.23	<.0001*
(h-0.28454)*(GroundDistance-0.79498)	0.0000406	7.2e-6	5.64	<.0001*
(HeadWind-10.7592)*(HeadWind-10.7592)	-2.207e-7	4.665e-8	-4.73	<.0001*
(InletMachNumber-0.4665)*(HeadWind-10.7592)	-9.219e-6	2.129e-6	-4.33	<.0001*
GroundDistance	7.5521e-7	2.115e-7	3.57	0.0004*
(InletMachNumber-0.4665)*(InletMachNumber-0.4665	-0.000216	0.000155	-1.39	0.1646
(GroundDistance-0.79498)*(HeadWind-10.7592)	-2.404e-7	2.315e-7	-1.04	0.3003

Figure G.4: Chord Length Goodness of Fit, 3.2 mm FOD

Particle radial location at impact fit and goodness of fit for two FOD sizes:

$$\text{Rimpact}_{fit,1.33mmFOD} = 0.065415445705254 + -0.156714564501262 * \text{h} + 0.000132668082703971 * \text{GroundDistance} + 0.64470128769476 * \text{InletMachNumber} + 0.00178339145613628 * \text{HeadWind} + (\text{h} - 0.285513648014925) * ((\text{h} - 0.285513648014925) * 0.00161247997422258) + (\text{h} - 0.285513648014925) * ((\text{GroundDistance} - 0.727088406885572) * 0.000201935239468427) + (\text{GroundDistance} - 0.727088406885572) * ((\text{GroundDistance} - 0.727088406885572) * -0.000362165459376822) + (\text{h} - 0.285513648014925) * ((\text{InletMachNumber} - 0.472574813666666) * 0.792303049785879) + (\text{GroundDistance} - 0.727088406885572) * ((\text{InletMachNumber} - 0.472574813666666) * 0.00131731731841904) + (\text{InletMachNumber} - 0.472574813666666) * ((\text{InletMachNumber} - 0.472574813666666) * -0.00732966024459297) + (\text{h} - 0.285513648014925) * ((\text{HeadWind} - 10.6542844452363) * -0.0000312086878645552) + (\text{GroundDistance} - 0.727088406885572) * ((\text{HeadWind} - 10.6542844452363) * -0.000003242570130508) + (\text{InletMachNumber} - 0.472574813666666) * ((\text{HeadWind} - 10.6542844452363) * -0.0000516841213850941) + (\text{HeadWind} - 10.6542844452363) * ((\text{HeadWind} - 10.6542844452363) * -0.0000598168644780257) \quad \text{(G.5)}$$



Summary of Fit

RSquare	1
RSquare Adj	1
Root Mean Square Error	4.158e-6
Mean of Response	0.344367
Observations (or Sum Wgts)	402

Analysis of Variance

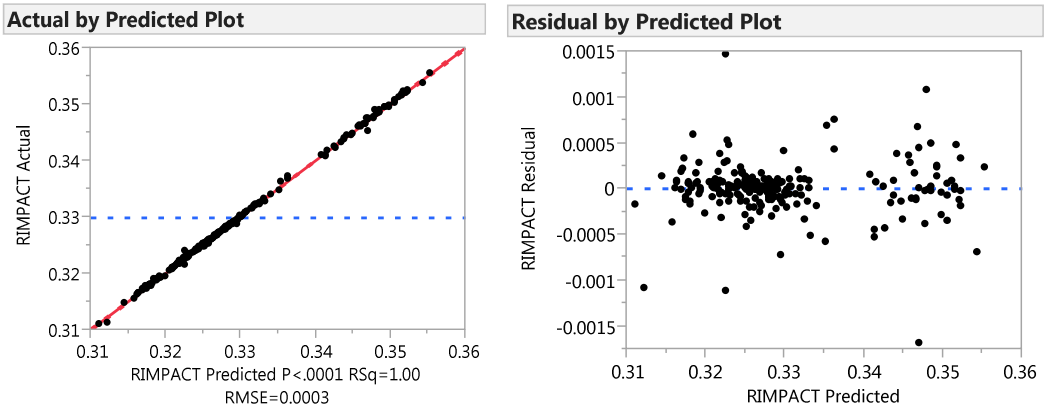
Source	DF	Sum of Squares	Mean Square	F Ratio
Model	14	0.05360933	0.003829	2.214e+8
Error	387	6.69194e-9	1.73e-11	**Prob > F**
C. Total	401	0.05360933		<.0001*

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
InletMachNumber	0.6447013	1.366e-5	47182	<.0001*
h	-0.156715	8.036e-6	-19501	<.0001*
HeadWind	0.0017834	2.681e-7	6652.5	<.0001*
(h-0.28551)*(InletMachNumber-0.47257)	0.792303	0.000495	1601.7	<.0001*
(HeadWind-10.6543)*(HeadWind-10.6543)	-5.982e-5	3.794e-7	-157.7	<.0001*
GroundDistance	0.0001327	1.465e-6	90.55	<.0001*
(GroundDistance-0.72709)*(GroundDistance-0.72709)	-0.000362	1.032e-5	-35.09	<.0001*
(GroundDistance-0.72709)*(InletMachNumber-0.47257)	0.0013173	0.000087	15.16	<.0001*
(InletMachNumber-0.47257)*(InletMachNumber-0.47257	-0.00733	0.001047	-7.00	<.0001*
(h-0.28551)*(h-0.28551)	0.0016125	0.000329	4.91	<.0001*
(h-0.28551)*(GroundDistance-0.72709)	0.0002019	5.36e-5	3.77	0.0002*
(InletMachNumber-0.47257)*(HeadWind-10.6543)	-5.168e-5	1.626e-5	-3.18	0.0016*
(h-0.28551)*(HeadWind-10.6543)	-3.121e-5	1.025e-5	-3.04	0.0025*
(GroundDistance-0.72709)*(HeadWind-10.6543)	-3.243e-6	1.85e-6	-1.75	0.0805

Figure G.5: Radial Impact Location Goodness of Fit, 1.33 mm FOD

$$\text{Rimpact}_{fit,3.2mmFOD} = 0.0478309777989539 + -0.111274961291789 * \text{h} + 0.000564033629471426 * \text{GroundDistance} + 0.630154293058168 * \text{InletMachNumber} + 0.00177464820018241 * \text{HeadWind} + (\text{h} - 0.284538338355556) * ((\text{h} - 0.284538338355556) * 0.207834798123669) + (\text{h} - 0.284538338355556) * ((\text{GroundDistance} - 0.794982760546667) * 0.0303413475087664) + (\text{GroundDistance} - 0.794982760546667) * ((\text{GroundDistance} - 0.794982760546667) * 0.00885091883909214) + (\text{h} - 0.284538338355556) * ((\text{InletMachNumber} - 0.466504282097778) * -0.304192130693951) + (\text{GroundDistance} - 0.794982760546667) * ((\text{InletMachNumber} - 0.466504282097778) * -0.0648212606492402) + (\text{InletMachNumber} - 0.466504282097778) * ((\text{InletMachNumber} - 0.466504282097778) * -0.161117138503253) + (\text{h} - 0.284538338355556) * ((\text{HeadWind} - 10.7592392263289) * 0.0100849844699348) + (\text{GroundDistance} - 0.794982760546667) * ((\text{HeadWind} - 10.7592392263289) * -0.000179596191982269) + (\text{InletMachNumber} - 0.466504282097778) * ((\text{HeadWind} - 10.7592392263289) * -0.00688575342193288) + (\text{HeadWind} - 10.7592392263289) * ((\text{HeadWind} - 10.7592392263289) * -0.000164871098217123) \quad \text{(G.6)}$$



Summary of Fit

RSquare	0.999224
RSquare Adj	0.999172
Root Mean Square Error	0.000296
Mean of Response	0.329845
Observations (or Sum Wgts)	225

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	14	0.02364514	0.001689	19304.86
Error	210	0.00001837	8.749e-8	Prob > F
C. Total	224	0.02366352		<.0001*

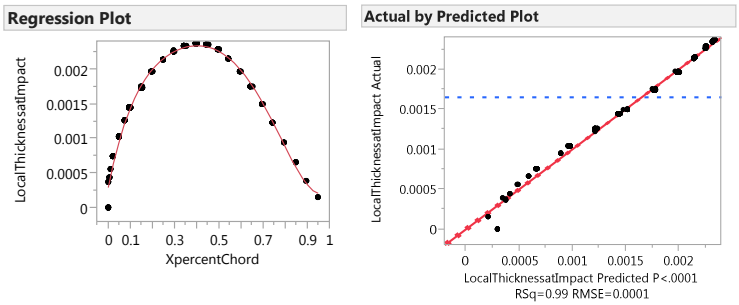
Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
InletMachNumber	0.6301543	0.002424	259.93	<.0001*
h	-0.111275	0.000799	-139.2	<.0001*
HeadWind	0.0017746	2.522e-5	70.35	<.0001*
(h-0.28454)*(HeadWind-10.7592)	0.010085	0.000945	10.67	<.0001*
(GroundDistance-0.79498)*(GroundDistance-0.79498)	0.0088509	0.001109	7.98	<.0001*
(GroundDistance-0.79498)*(InletMachNumber-0.4665)	-0.064821	0.008565	-7.57	<.0001*
(h-0.28454)*(h-0.28454)	0.2078348	0.032695	6.36	<.0001*
(h-0.28454)*(InletMachNumber-0.4665)	-0.304192	0.048843	-6.23	<.0001*
(h-0.28454)*(GroundDistance-0.79498)	0.0303413	0.00538	5.64	<.0001*
(HeadWind-10.7592)*(HeadWind-10.7592)	-0.000165	3.486e-5	-4.73	<.0001*
(InletMachNumber-0.4665)*(HeadWind-10.7592)	-0.006886	0.00159	-4.33	<.0001*
GroundDistance	0.000564	0.000158	3.57	0.0004*
(InletMachNumber-0.4665)*(InletMachNumber-0.4665	-0.161117	0.115532	-1.39	0.1646
(GroundDistance-0.79498)*(HeadWind-10.7592)	-0.00018	0.000173	-1.04	0.3003

Figure G.6: Radial Impact Location Goodness of Fit, 3.2 mm FOD

Local airfoil thickness at impact fit and goodness of fit for two FOD sizes:

$$\begin{aligned}\text{LocalThicknessatImpact, fit, 1.33mmFOD} = {} & 0.00186625343042413 + \\ & 0.00132154137591184 * \text{XpercentChord} + (\text{XpercentChord} - \\ & 0.317678693072139) * ((\text{XpercentChord} - 0.317678693072139) * \\ & -0.0080516775782884) + (\text{XpercentChord} - 0.317678693072139) * \\ & ((\text{XpercentChord} - 0.317678693072139) * ((\text{XpercentChord} - \\ & 0.317678693072139) * 0.00548564206578457)) + (\text{XpercentChord} - \\ & 0.317678693072139) * ((\text{XpercentChord} - 0.317678693072139) * \\ & ((\text{XpercentChord} - 0.317678693072139) * ((\text{XpercentChord} - \\ & 0.317678693072139) * -0.0402999953775497))) + (\text{XpercentChord} - \\ & 0.317678693072139) * ((\text{XpercentChord} - 0.317678693072139) * \\ & ((\text{XpercentChord} - 0.317678693072139) * ((\text{XpercentChord} - \\ & 0.317678693072139) * ((\text{XpercentChord} - 0.317678693072139) * \\ & 0.0530674615909787)))) \end{aligned} \quad \text{(G.7)}$$



Summary of Fit

RSquare	0.992595
RSquare Adj	0.992502
Root Mean Square Error	5.49e-5
Mean of Response	0.001649
Observations (or Sum Wgts)	402

Analysis of Variance

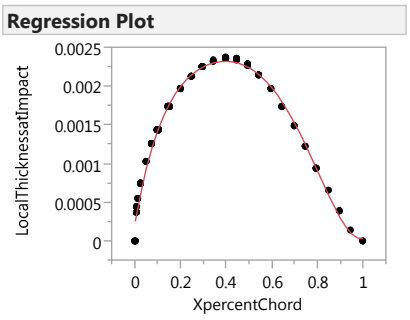
Source	DF	Sum of Squares	Mean Square	F Ratio
Model	5	0.00015996	0.000032	10616.37
Error	396	0.00000119	3.013e-9	Prob > F
C. Total	401	0.00016116		<.0001*

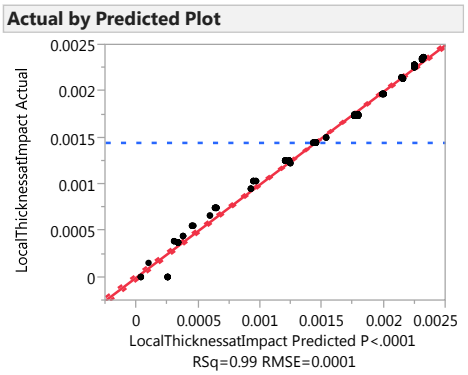
Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	0.0018663	1.256e-5	148.60	<.0001*
XpercentChord	0.0013215	3.462e-5	38.18	<.0001*
(XpercentChord-0.31768)*(XpercentChord-0.31768)	-0.008052	0.000248	-32.52	<.0001*
(XpercentChord-0.31768)*(XpercentChord-0.31768)*(XpercentChord-0.31768)	0.0054856	0.000482	11.37	<.0001*
(XpercentChord-0.31768)*(XpercentChord-0.31768)*(XpercentChord-0.31768)*(XpercentChord-0.31768)	-0.0403	0.002139	-18.84	<.0001*
(XpercentChord-0.31768)*(XpercentChord-0.31768)*(XpercentChord-0.31768)*(XpercentChord-0.31768)*(XpercentChord-0.31768	0.0530675	0.003056	17.37	<.0001*

Figure G.7: Local Airfoil Thickness at Impact Goodness of Fit, 1.33 mm FOD

$$\text{LocalThicknessatImpact} = 0.00167764838585954 + 0.00196207927772939 * \text{XpercentChord} + (\text{XpercentChord} - 0.272987192546667) * ((\text{XpercentChord} - 0.272987192546667) * -0.00969194810438932) + (\text{XpercentChord} - 0.272987192546667) * ((\text{XpercentChord} - 0.272987192546667) * ((\text{XpercentChord} - 0.272987192546667) * 0.0163865776570472)) + (\text{XpercentChord} - 0.272987192546667) * ((\text{XpercentChord} - 0.272987192546667) * ((\text{XpercentChord} - 0.272987192546667) * ((\text{XpercentChord} - 0.272987192546667) * -0.0519913018899497))) + (\text{XpercentChord} - 0.272987192546667) * ((\text{XpercentChord} - 0.272987192546667) * ((\text{XpercentChord} - 0.272987192546667) * ((\text{XpercentChord} - 0.272987192546667) * ((\text{XpercentChord} - 0.272987192546667) * 0.0480118431095079)))) \quad \text{(G.8)}$$





Summary of Fit

RSquare	0.991396
RSquare Adj	0.9912
Root Mean Square Error	0.000069
Mean of Response	0.001445
Observations (or Sum Wgts)	225

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	5	0.00012029	0.000024	5046.950
Error	219	0.00000104	4.767e-9	Prob > F
C. Total	224	0.00012133		<.0001*

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	0.0016776	1.744e-5	96.20	<.0001*
XpercentChord	0.0019621	0.000063	31.11	<.0001*
(XpercentChord-0.27299)*(XpercentChord-0.27299)	-0.009692	0.000387	-25.05	<.0001*
(XpercentChord-0.27299)*(XpercentChord-0.27299)*(XpercentChord-0.27299)	0.0163866	0.000806	20.34	<.0001*
(XpercentChord-0.27299)*(XpercentChord-0.27299)*(XpercentChord-0.27299)*(XpercentChord-0.27299)	-0.051991	0.003825	-13.59	<.0001*
(XpercentChord-0.27299)*(XpercentChord-0.27299)*(XpercentChord-0.27299)*(XpercentChord-0.27299)*(XpercentChord-0.27299	0.0480118	0.003781	12.70	<.0001*

Figure G.8: Local Airfoil Thickness at Impact Goodness of Fit, 3.2 mm FOD

Beta distributions for non-dimensional chord and depth of penetration at impact for both FOD sizes:

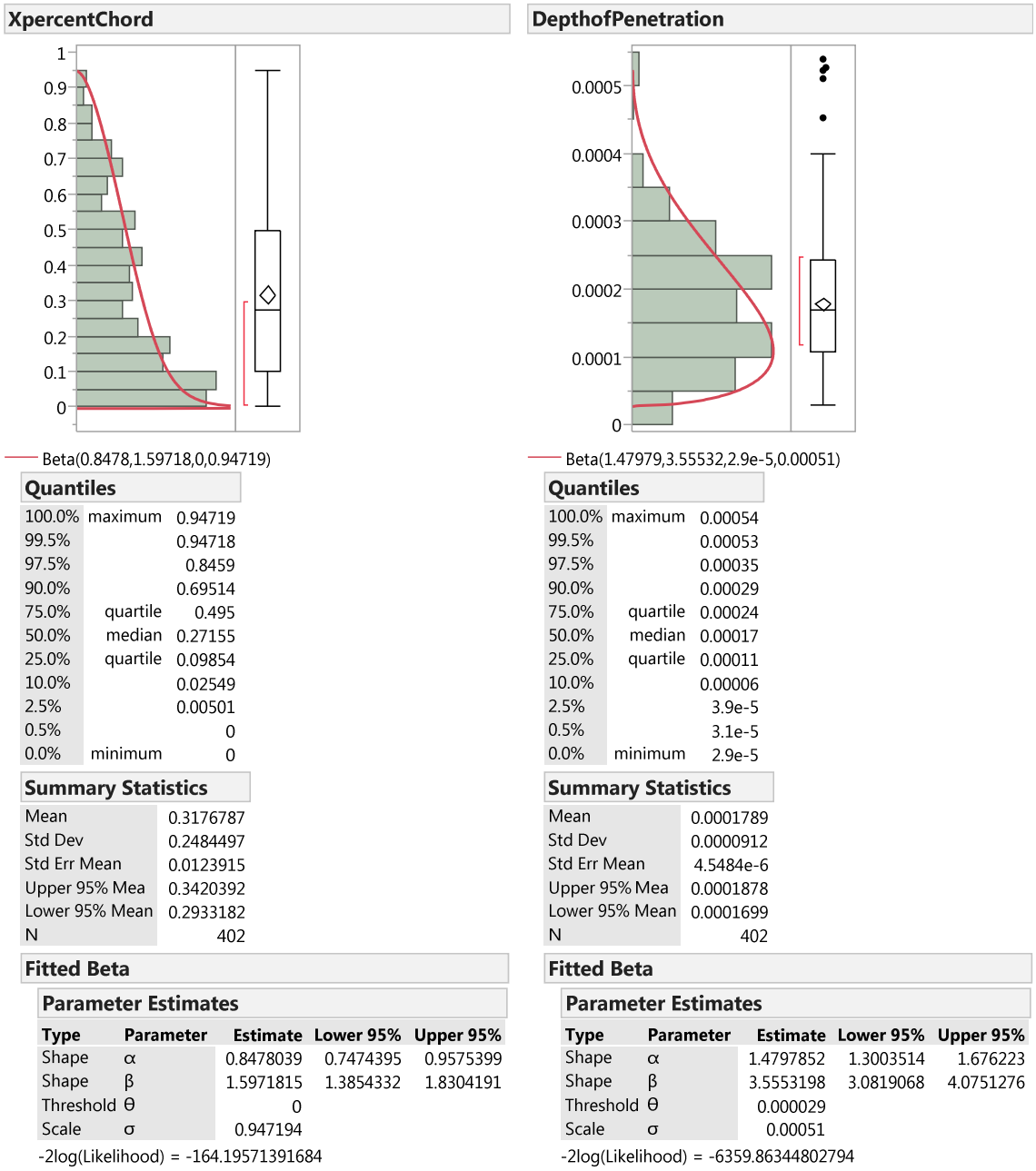
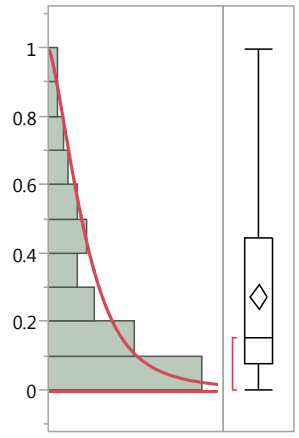


Figure G.9: Non-dim Chord and Depth of Penetration Beta Dist., 1.33 mm FOD

XpercentChord



Beta(0.66125,1.75476,0,0.99634)

Quantiles

100.0%	maximum	0.99634
99.5%		0.99634
97.5%		0.91375
90.0%		0.66505
75.0%	quartile	0.44521
50.0%	median	0.15215
25.0%	quartile	0.07388
10.0%		0.00503
2.5%		0
0.5%		0
0.0%	minimum	0

Summary Statistics

Mean	0.2729872
Std Dev	0.2639776
Std Err Mean	0.0175985
Upper 95% Mea	0.307667
Lower 95% Mean	0.2383074
N	225

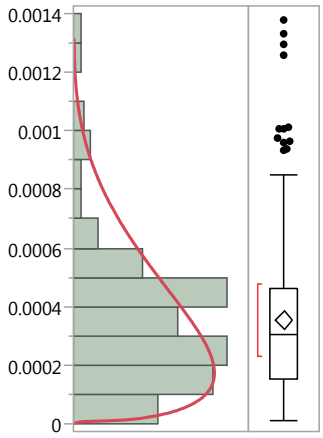
Fitted Beta

Parameter Estimates

Type	Parameter	Estimate	Lower 95%	Upper 95%
Shape	α	0.6612481	0.5594287	0.7758709
Shape	β	1.7547565	1.4302811	2.1267635
Threshold	θ	0		
Scale	σ	0.9963398		

-2log(Likelihood) = -157.146099118283

DepthofPenetration



Beta(1.4217,4.04856,0.00001,0.00137)

Quantiles

100.0%	maximum	0.00138
99.5%		0.00137
97.5%		0.00101
90.0%		0.00061
75.0%	quartile	0.00046
50.0%	median	0.0003
25.0%	quartile	0.00015
10.0%		0.0001
2.5%		5.18e-5
0.5%		1.17e-5
0.0%	minimum	0.00001

Summary Statistics

Mean	0.0003553
Std Dev	0.0002443
Std Err Mean	1.6287e-5
Upper 95% Mea	0.0003874
Lower 95% Mean	0.0003232
N	225

Fitted Beta

Parameter Estimates

Type	Parameter	Estimate	Lower 95%	Upper 95%
Shape	α	1.4216988	1.1954665	1.677027
Shape	β	4.0485604	3.3308299	4.8622593
Threshold	θ	1.0313e-5		
Scale	σ	0.0013681		

-2log(Likelihood) = -3147.03093074972

Figure G.10: Non-dim Chord and Depth of Penetration Beta Dist., 3.2 mm FOD

APPENDIX H: PARTICLE IMPACT OUTPUT DATA

This appendix presents the screening test plots for radial location, non-dimensional chord-wise location on the airfoil, impact side, impact velocity and depth of penetration of particles that impacted the 1st HPC rotor blades - refer to Figure 5-16 as a rubric to discern statistically significant factors from the figures presented in this appendix.

Term	Contrast	Lenth t-Ratio	Individual p-Value
Mi	0.011425	2161.44	<.0001*
ParticleSize	-0.005010	-947.72	<.0001*
h	-0.003682	-696.56	<.0001*
U∞	0.001440	272.41	<.0001*
GroundDistance	0.000051	9.64	<.0001*
Mi*Mi	7.891e-6	1.49	0.1387
Mi*ParticleSize	-0.000149	-28.10	<.0001*
Mi*h	0.000074	14.01	<.0001*
ParticleSize*h	0.000585	110.58	<.0001*
h*h	0.000048	9.14	<.0001*
Mi*U∞	-0.000056	-10.66	<.0001*
ParticleSize*U∞	-9.142e-6	-1.73	0.0883
h*U∞	0.000069	13.14	<.0001*
U∞*U∞	-0.000043	-8.10	<.0001*
Mi*GroundDistance	-0.000053	-10.05	<.0001*
ParticleSize*GroundDistance	-0.000016	-2.94	0.0048*
h*GroundDistance	0.000030	5.60	<.0001*
U∞*GroundDistance	7.8717e-7	0.15	0.8772
GroundDistance*GroundDistance	0.000058	10.97	<.0001*
Mi*Mi*Mi	-4.739e-6	-0.90	0.3777
Mi*Mi*ParticleSize	0.000024	4.63	<.0001*
Mi*Mi*h	-0.000016	-3.07	0.0036*
Mi*ParticleSize*h	-0.000210	-39.72	<.0001*
Mi*h*h	-0.000037	-7.01	<.0001*
ParticleSize*h*h	0.000071	13.51	<.0001*
h*h*h	0.000013	2.49	0.0149*
Mi*Mi*U∞	0.000012	2.20	0.0304*
Mi*ParticleSize*U∞	-0.000050	-9.39	<.0001*
Mi*h*U∞	-0.000042	-7.86	<.0001*
ParticleSize*h*U∞	0.000091	17.19	<.0001*
h*h*U∞	0.000034	6.35	<.0001*
Mi*U∞*U∞	2.8144e-6	0.53	0.5980
ParticleSize*U∞*U∞	-0.000024	-4.47	<.0001*
h*U∞*U∞	-0.000015	-2.76	0.0076*
U∞*U∞*U∞	0.000011	2.11	0.0390*
Mi*Mi*GroundDistance	0.000013	2.51	0.0143*
Mi*ParticleSize*GroundDistance	-0.000068	-12.89	<.0001*
Mi*h*GroundDistance	0.000039	7.42	<.0001*
ParticleSize*h*GroundDistance	0.000066	12.39	<.0001*
h*h*GroundDistance	0.000015	2.78	0.0070*
Mi*U∞*GroundDistance	0.000039	7.45	<.0001*
ParticleSize*U∞*GroundDistance	1.4181e-6	0.27	0.7827
h*U∞*GroundDistance	0.000083	15.68	<.0001*
U∞*U∞*GroundDistance	1.9983e-6	0.38	0.7005
Mi*GroundDistance*GroundDistance	-0.000011	-2.17	0.0336*
ParticleSize*GroundDistance*GroundDistance	0.000079	15.02	<.0001*
h*GroundDistance*GroundDistance	1.363e-6	0.26	0.7920
U∞*GroundDistance*GroundDistance	0.000013	2.44	0.0164*
GroundDistance*GroundDistance*GroundDistance	4.2882e-7	0.08	0.9307

Figure H.1: Factors vs. Radial Impact Location

Term	Contrast	Lenth t-Ratio	Individual p-Value
GroundDistance	-0.028514	-2.72	0.0082*
ParticleSize	-0.015526	-1.48	0.1418
Mi	-0.002793	-0.27	0.8005
h	-0.002156	-0.21	0.8434
U∞	-0.000227	-0.02	0.9846
GroundDistance*GroundDistance	0.006966	0.67	0.5123
GroundDistance*ParticleSize	0.006995	0.67	0.5103
GroundDistance*Mi	-0.012560	-1.20	0.2353
ParticleSize*Mi	-0.004060	-0.39	0.7087
Mi*Mi	-0.012186	-1.16	0.2494
GroundDistance*h	0.008028	0.77	0.4481
ParticleSize*h	-0.020694	-1.98	0.0530
Mi*h	-0.011574	-1.11	0.2726
h*h	-0.000398	-0.04	0.9720
GroundDistance*U∞	0.010638	1.02	0.3128
ParticleSize*U∞	-5.066e-6	-0.00	0.9997
Mi*U∞	0.002623	0.25	0.8116
h*U∞	0.013879	1.33	0.1922
U∞*U∞	0.015333	1.46	0.1474
GroundDistance*GroundDistance*GroundDistance	0.008399	0.80	0.4281
GroundDistance*GroundDistance*ParticleSize	0.010666	1.02	0.3116
GroundDistance*GroundDistance*Mi	0.001222	0.12	0.9093
GroundDistance*ParticleSize*Mi	-0.005420	-0.52	0.6144
GroundDistance*Mi*Mi	0.006312	0.60	0.5544
ParticleSize*Mi*Mi	0.001293	0.12	0.9049
Mi*Mi*Mi	0.001522	0.15	0.8883
GroundDistance*GroundDistance*h	-0.009120	-0.87	0.3892
GroundDistance*ParticleSize*h	-0.013167	-1.26	0.2147
GroundDistance*Mi*h	-0.004692	-0.45	0.6632
ParticleSize*Mi*h	-0.001271	-0.12	0.9062
Mi*Mi*h	-0.013602	-1.30	0.2009
GroundDistance*h*h	0.012451	1.19	0.2396
ParticleSize*h*h	-0.010752	-1.03	0.3079
Mi*h*h	-0.012485	-1.19	0.2385
h*h*h	-0.002136	-0.20	0.8452
GroundDistance*GroundDistance*U∞	-0.002227	-0.21	0.8394
GroundDistance*ParticleSize*U∞	0.000928	0.09	0.9319
GroundDistance*Mi*U∞	0.011045	1.05	0.2964
ParticleSize*Mi*U∞	-0.008792	-0.84	0.4070
Mi*Mi*U∞	0.008531	0.81	0.4205
GroundDistance*h*U∞	-0.002155	-0.21	0.8434
ParticleSize*h*U∞	-0.010540	-1.01	0.3169
Mi*h*U∞	0.006745	0.64	0.5268
h*h*U∞	-0.010038	-0.96	0.3424
GroundDistance*U∞*U∞	-0.007097	-0.68	0.5032
ParticleSize*U∞*U∞	0.017845	1.70	0.0908
Mi*U∞*U∞	0.005439	0.52	0.6129
h*U∞*U∞	-0.001994	-0.19	0.8556
U∞*U∞*U∞	-0.000920	-0.09	0.9323

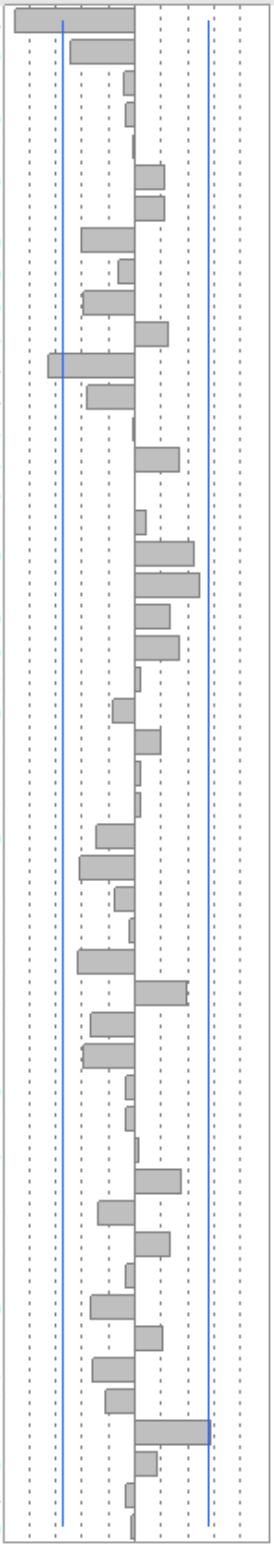
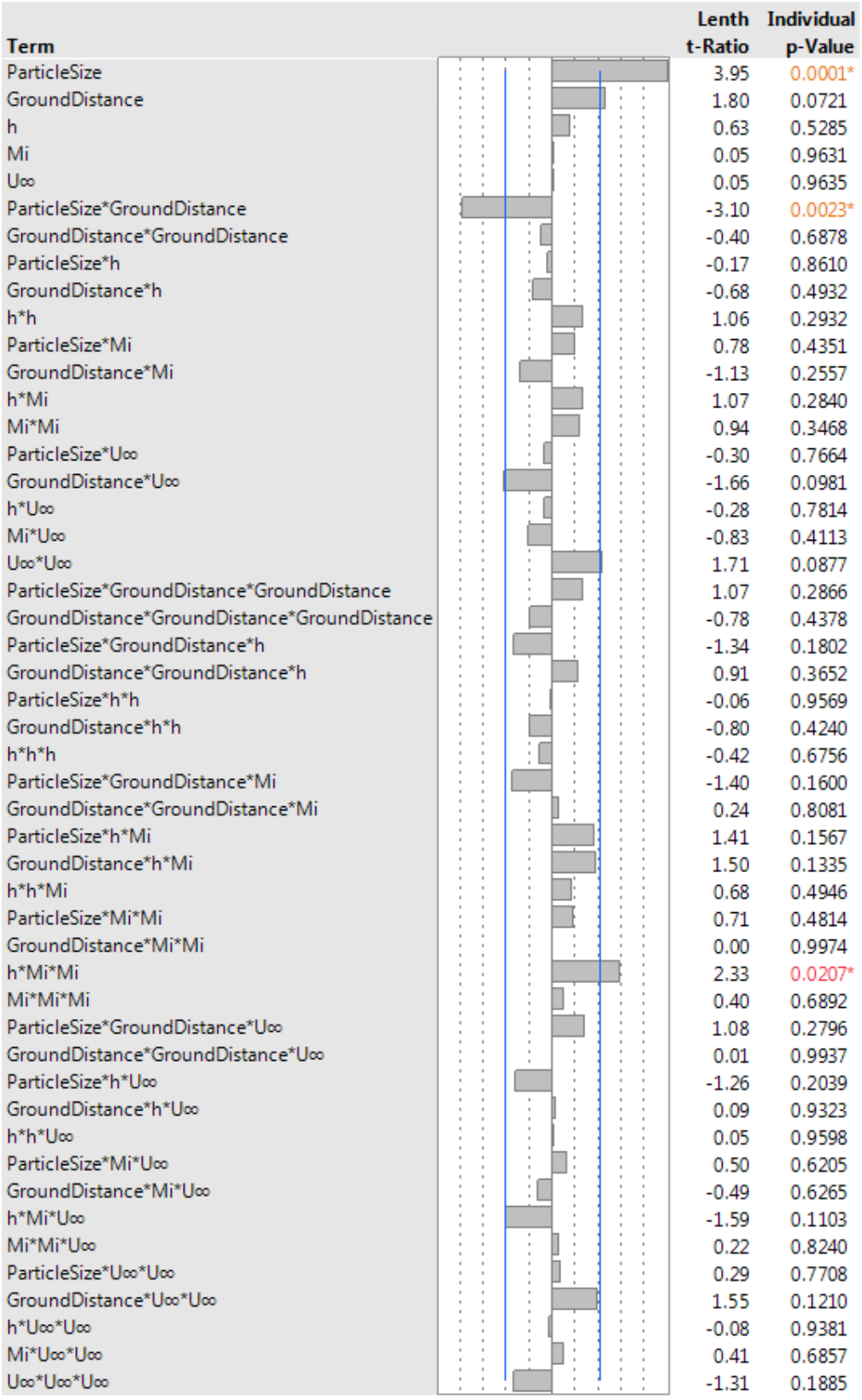


Figure H.2: Factors vs. Non-dim Chord Impact Location



Term	Lenth t-Ratio	Individual p-Value
ParticleSize	3.95	0.0001*
GroundDistance	1.80	0.0721
h	0.63	0.5285
Mi	0.05	0.9631
U∞	0.05	0.9635
ParticleSize*GroundDistance	-3.10	0.0023*
GroundDistance*GroundDistance	-0.40	0.6878
ParticleSize*h	-0.17	0.8610
GroundDistance*h	-0.68	0.4932
h*h	1.06	0.2932
ParticleSize*Mi	0.78	0.4351
GroundDistance*Mi	-1.13	0.2557
h*Mi	1.07	0.2840
Mi*Mi	0.94	0.3468
ParticleSize*U∞	-0.30	0.7664
GroundDistance*U∞	-1.66	0.0981
h*U∞	-0.28	0.7814
Mi*U∞	-0.83	0.4113
U∞*U∞	1.71	0.0877
ParticleSize*GroundDistance*GroundDistance	1.07	0.2866
GroundDistance*GroundDistance*GroundDistance	-0.78	0.4378
ParticleSize*GroundDistance*h	-1.34	0.1802
GroundDistance*GroundDistance*h	0.91	0.3652
ParticleSize*h*h	-0.06	0.9569
GroundDistance*h*h	-0.80	0.4240
h*h*h	-0.42	0.6756
ParticleSize*GroundDistance*Mi	-1.40	0.1600
GroundDistance*GroundDistance*Mi	0.24	0.8081
ParticleSize*h*Mi	1.41	0.1567
GroundDistance*h*Mi	1.50	0.1335
h*h*Mi	0.68	0.4946
ParticleSize*Mi*Mi	0.71	0.4814
GroundDistance*Mi*Mi	0.00	0.9974
h*Mi*Mi	2.33	0.0207*
Mi*Mi*Mi	0.40	0.6892
ParticleSize*GroundDistance*U∞	1.08	0.2796
GroundDistance*GroundDistance*U∞	0.01	0.9937
ParticleSize*h*U∞	-1.26	0.2039
GroundDistance*h*U∞	0.09	0.9323
h*h*U∞	0.05	0.9598
ParticleSize*Mi*U∞	0.50	0.6205
GroundDistance*Mi*U∞	-0.49	0.6265
h*Mi*U∞	-1.59	0.1103
Mi*Mi*U∞	0.22	0.8240
ParticleSize*U∞*U∞	0.29	0.7708
GroundDistance*U∞*U∞	1.55	0.1210
h*U∞*U∞	-0.08	0.9381
Mi*U∞*U∞	0.41	0.6857
U∞*U∞*U∞	-1.31	0.1885

Figure H.3: Factors vs. Impact Velocity

Term	Lenth t-Ratio	Individual p-Value
h	-0.73	0.4652
Mi	0.58	0.5610
GroundDistance	0.37	0.7077
U∞	0.03	0.9751
h*h	-1.31	0.1936
h*Mi	-0.40	0.6890
Mi*Mi	0.37	0.7111
h*GroundDistance	-0.27	0.7869
Mi*GroundDistance	1.15	0.2532
GroundDistance*GroundDistance	1.04	0.2978
h*U∞	-1.24	0.2159
Mi*U∞	-0.82	0.4083
GroundDistance*U∞	1.27	0.2076
U∞*U∞	-2.19	0.0276*
h*h*h	0.77	0.4371
h*h*Mi	0.67	0.5010
h*Mi*Mi	0.38	0.7009
Mi*Mi*Mi	0.58	0.5586
h*h*GroundDistance	-0.35	0.7259
h*Mi*GroundDistance	-2.23	0.0252*
Mi*Mi*GroundDistance	0.02	0.9850
h*GroundDistance*GroundDistance	0.61	0.5379
Mi*GroundDistance*GroundDistance	-0.40	0.6833
GroundDistance*GroundDistance*GroundDistance	-0.67	0.4992
h*h*U∞	1.10	0.2715
h*Mi*U∞	0.55	0.5767
Mi*Mi*U∞	-0.92	0.3533
h*GroundDistance*U∞	0.54	0.5830
Mi*GroundDistance*U∞	0.28	0.7755
GroundDistance*GroundDistance*U∞	-0.33	0.7407
h*U∞*U∞	-0.52	0.6013
Mi*U∞*U∞	-1.38	0.1667
GroundDistance*U∞*U∞	-1.11	0.2675
U∞*U∞*U∞	1.47	0.1430

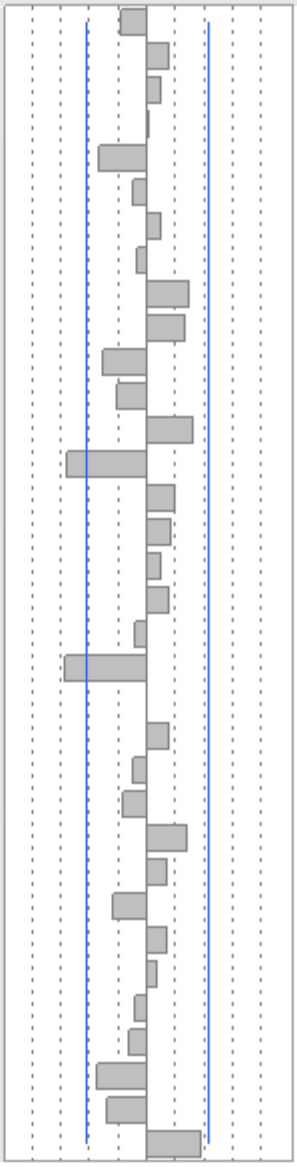
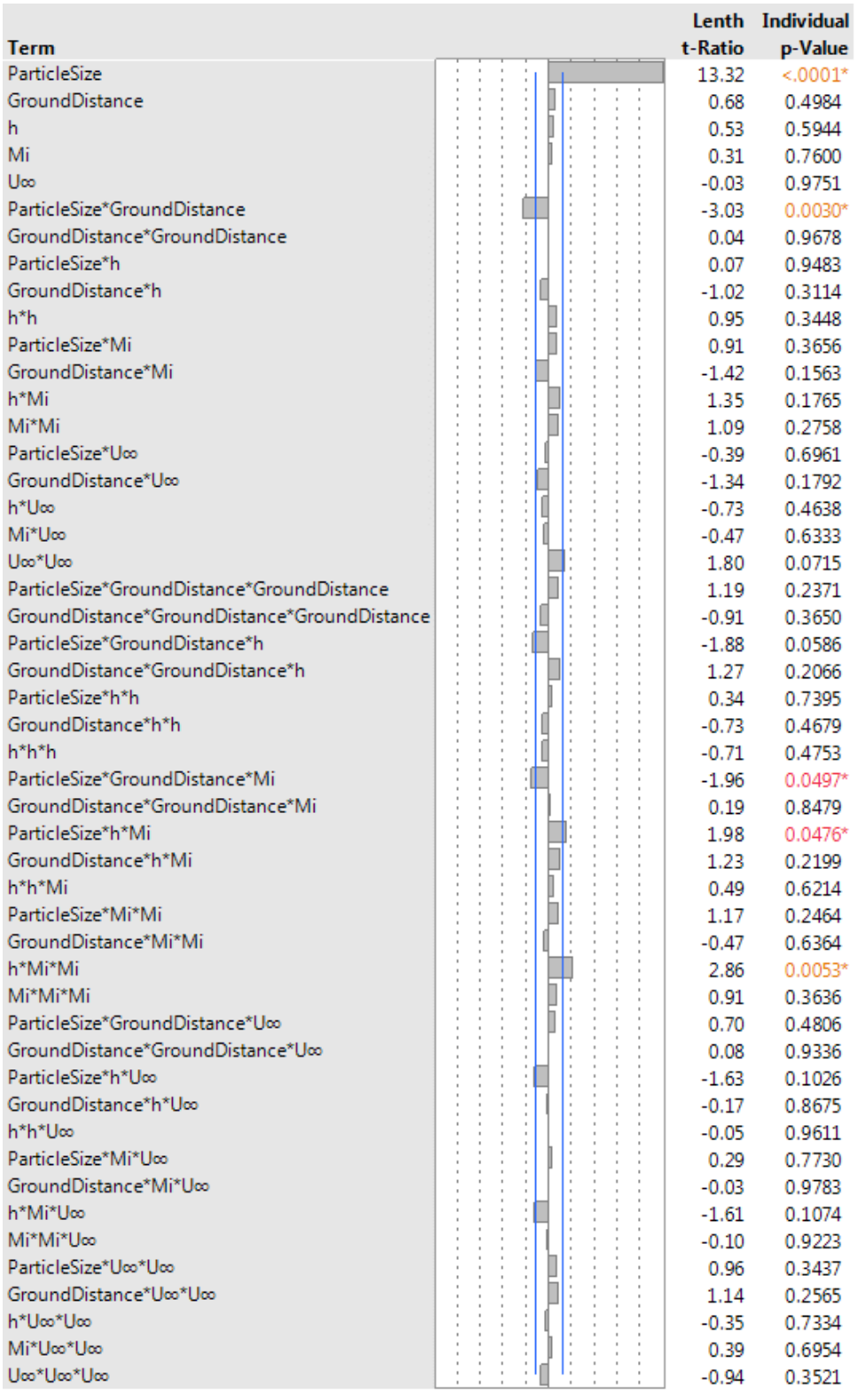


Figure H.4: Factors vs. Impact Side

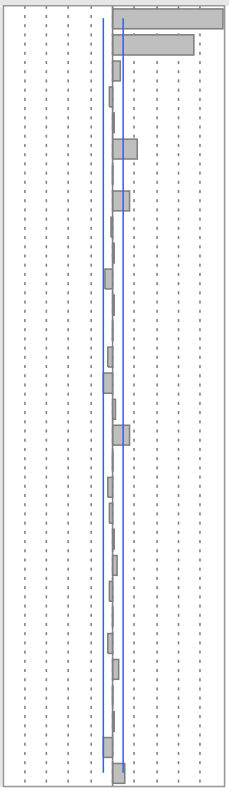


Term	Lenth t-Ratio	Individual p-Value
ParticleSize	13.32	<.0001*
GroundDistance	0.68	0.4984
h	0.53	0.5944
Mi	0.31	0.7600
U∞	-0.03	0.9751
ParticleSize*GroundDistance	-3.03	0.0030*
GroundDistance*GroundDistance	0.04	0.9678
ParticleSize*h	0.07	0.9483
GroundDistance*h	-1.02	0.3114
h*h	0.95	0.3448
ParticleSize*Mi	0.91	0.3656
GroundDistance*Mi	-1.42	0.1563
h*Mi	1.35	0.1765
Mi*Mi	1.09	0.2758
ParticleSize*U∞	-0.39	0.6961
GroundDistance*U∞	-1.34	0.1792
h*U∞	-0.73	0.4638
Mi*U∞	-0.47	0.6333
U∞*U∞	1.80	0.0715
ParticleSize*GroundDistance*GroundDistance	1.19	0.2371
GroundDistance*GroundDistance*GroundDistance	-0.91	0.3650
ParticleSize*GroundDistance*h	-1.88	0.0586
GroundDistance*GroundDistance*h	1.27	0.2066
ParticleSize*h*h	0.34	0.7395
GroundDistance*h*h	-0.73	0.4679
h*h*h	-0.71	0.4753
ParticleSize*GroundDistance*Mi	-1.96	0.0497*
GroundDistance*GroundDistance*Mi	0.19	0.8479
ParticleSize*h*Mi	1.98	0.0476*
GroundDistance*h*Mi	1.23	0.2199
h*h*Mi	0.49	0.6214
ParticleSize*Mi*Mi	1.17	0.2464
GroundDistance*Mi*Mi	-0.47	0.6364
h*Mi*Mi	2.86	0.0053*
Mi*Mi*Mi	0.91	0.3636
ParticleSize*GroundDistance*U∞	0.70	0.4806
GroundDistance*GroundDistance*U∞	0.08	0.9336
ParticleSize*h*U∞	-1.63	0.1026
GroundDistance*h*U∞	-0.17	0.8675
h*h*U∞	-0.05	0.9611
ParticleSize*Mi*U∞	0.29	0.7730
GroundDistance*Mi*U∞	-0.03	0.9783
h*Mi*U∞	-1.61	0.1074
Mi*Mi*U∞	-0.10	0.9223
ParticleSize*U∞*U∞	0.96	0.3437
GroundDistance*U∞*U∞	1.14	0.2565
h*U∞*U∞	-0.35	0.7334
Mi*U∞*U∞	0.39	0.6954
U∞*U∞*U∞	-0.94	0.3521

Figure H.5: Factors vs. Impact Depth of Penetration

APPENDIX I: LCC, LRU REPLACEMENT/REPAIR OUTPUT DATA

This appendix presents the screening test plots for LCCs, LRU replacements and repairs vs. factors for - refer to Figure 5-16 as a rubric to discern statistically significant factors from the figures presented in this appendix.



Term	Lenth t-Ratio	Individual p-Value
ParticleDiameter	17.31	<.0001*
DTAEngineInspections	12.83	<.0001*
LSP	1.18	0.2325
NumberRunwayVisualInspections	-0.49	0.6192
RunwaySweeps	0.17	0.8571
ParticleDiameter*DTAEngineInspections	3.91	0.0003*
ParticleDiameter*LSP	0.05	0.9580
DTAEngineInspections*LSP	2.73	0.0061*
ParticleDiameter*NumberRunwayVisualInspections	-0.21	0.8292
DTAEngineInspections*NumberRunwayVisualInspections	0.15	0.8701
LSP*NumberRunwayVisualInspections	-1.21	0.2218
NumberRunwayVisualInspections*NumberRunwayVisualInspections	0.20	0.8364
ParticleDiameter*RunwaySweeps	-0.11	0.8995
DTAEngineInspections*RunwaySweeps	-0.68	0.4927
LSP*RunwaySweeps	-1.39	0.1607
NumberRunwayVisualInspections*RunwaySweeps	0.50	0.6160
ParticleDiameter*DTAEngineInspections*LSP	2.77	0.0055*
ParticleDiameter*DTAEngineInspections*NumberRunwayVisualInspections	0.01	0.9936
ParticleDiameter*LSP*NumberRunwayVisualInspections	-0.65	0.5137
DTAEngineInspections*LSP*NumberRunwayVisualInspections	-0.49	0.6208
ParticleDiameter*NumberRunwayVisualInspections*NumberRunwayVisualInspections	0.22	0.8163
DTAEngineInspections*NumberRunwayVisualInspections*NumberRunwayVisualInspections	0.83	0.3991
LSP*NumberRunwayVisualInspections*NumberRunwayVisualInspections	-0.43	0.6684
ParticleDiameter*DTAEngineInspections*RunwaySweeps	-0.08	0.9343
ParticleDiameter*LSP*RunwaySweeps	-0.76	0.4421
DTAEngineInspections*LSP*RunwaySweeps	1.07	0.2778
ParticleDiameter*NumberRunwayVisualInspections*RunwaySweeps	-0.05	0.9597
DTAEngineInspections*NumberRunwayVisualInspections*RunwaySweeps	0.25	0.7924
LSP*NumberRunwayVisualInspections*RunwaySweeps	-1.37	0.1683
NumberRunwayVisualInspections*NumberRunwayVisualInspections*RunwaySweeps	2.06	0.0409*

Figure I.1: Factors vs. LCCs, Runway Visual Inspections

Term	Lenth t-Ratio	Individual p-Value
ParticleDiameter	9.98	<.0001*
DTAEngineInspections	-7.29	<.0001*
LSP	-3.40	0.0010*
RunwaySweeps	0.49	0.6240
NumberRunwayVisualInspections	-0.38	0.7051
ParticleDiameter*DTAEngineInspections	-4.88	<.0001*
ParticleDiameter*LSP	-0.85	0.4011
DTAEngineInspections*LSP	2.35	0.0194*
ParticleDiameter*RunwaySweeps	0.31	0.7571
DTAEngineInspections*RunwaySweeps	-0.13	0.8899
LSP*RunwaySweeps	-1.75	0.0808
ParticleDiameter*NumberRunwayVisualInspections	-0.71	0.4835
DTAEngineInspections*NumberRunwayVisualInspections	0.11	0.9080
LSP*NumberRunwayVisualInspections	-0.73	0.4749
RunwaySweeps*NumberRunwayVisualInspections	0.55	0.5873
NumberRunwayVisualInspections*NumberRunwayVisualInspections	0.28	0.7801
ParticleDiameter*DTAEngineInspections*LSP	0.82	0.4189
ParticleDiameter*DTAEngineInspections*RunwaySweeps	0.13	0.8899
ParticleDiameter*LSP*RunwaySweeps	-0.74	0.4701
DTAEngineInspections*LSP*RunwaySweeps	1.16	0.2493
ParticleDiameter*DTAEngineInspections*NumberRunwayVisualInspections	0.22	0.8264
ParticleDiameter*LSP*NumberRunwayVisualInspections	-0.28	0.7857
DTAEngineInspections*LSP*NumberRunwayVisualInspections	0.03	0.9715
ParticleDiameter*RunwaySweeps*NumberRunwayVisualInspections	-0.11	0.9080
DTAEngineInspections*RunwaySweeps*NumberRunwayVisualInspections	0.05	0.9529
LSP*RunwaySweeps*NumberRunwayVisualInspections	-0.42	0.6792
ParticleDiameter*NumberRunwayVisualInspections*NumberRunwayVisualInspections	0.22	0.8248
DTAEngineInspections*NumberRunwayVisualInspections*NumberRunwayVisualInspections	0.44	0.6588
LSP*NumberRunwayVisualInspections*NumberRunwayVisualInspections	-0.54	0.5922
RunwaySweeps*NumberRunwayVisualInspections*NumberRunwayVisualInspections	0.63	0.5321

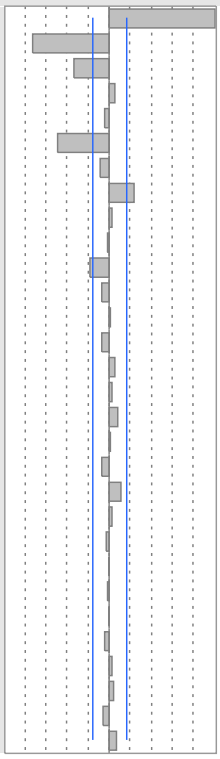


Figure I.2: Factors vs. LRU Replacements, Runway Visual Inspections

Term	Lenth t-Ratio	Individual p-Value
DTAEngineInspections	28.68	<.0001*
ParticleDiameter	20.74	<.0001*
RunwaySweeps	-0.51	0.6115
NumberRunwayVisualInspections	-0.17	0.8661
LSP	-0.06	0.9479
DTAEngineInspections*ParticleDiameter	20.74	<.0001*
DTAEngineInspections*RunwaySweeps	-0.51	0.6115
ParticleDiameter*RunwaySweeps	0.16	0.8743
DTAEngineInspections*NumberRunwayVisualInspections	-0.17	0.8661
ParticleDiameter*NumberRunwayVisualInspections	0.99	0.3161
RunwaySweeps*NumberRunwayVisualInspections	0.41	0.6843
NumberRunwayVisualInspections*NumberRunwayVisualInspections	-0.13	0.8985
DTAEngineInspections*LSP	-0.06	0.9479
ParticleDiameter*LSP	2.19	0.0287*
RunwaySweeps*LSP	1.31	0.1857
NumberRunwayVisualInspections*LSP	-1.35	0.1746
DTAEngineInspections*ParticleDiameter*RunwaySweeps	0.16	0.8743
DTAEngineInspections*ParticleDiameter*NumberRunwayVisualInspections	0.99	0.3161
DTAEngineInspections*RunwaySweeps*NumberRunwayVisualInspections	0.41	0.6843
ParticleDiameter*RunwaySweeps*NumberRunwayVisualInspections	0.31	0.7588
DTAEngineInspections*NumberRunwayVisualInspections*NumberRunwayVisualInspections	-0.13	0.8985
ParticleDiameter*NumberRunwayVisualInspections*NumberRunwayVisualInspections	0.04	0.9654
RunwaySweeps*NumberRunwayVisualInspections*NumberRunwayVisualInspections	1.61	0.1041
DTAEngineInspections*ParticleDiameter*LSP	2.19	0.0287*
DTAEngineInspections*RunwaySweeps*LSP	1.31	0.1857
ParticleDiameter*RunwaySweeps*LSP	0.69	0.4915
DTAEngineInspections*NumberRunwayVisualInspections*LSP	-1.35	0.1746
ParticleDiameter*NumberRunwayVisualInspections*LSP	-1.21	0.2220
RunwaySweeps*NumberRunwayVisualInspections*LSP	-2.29	0.0226*
NumberRunwayVisualInspections*NumberRunwayVisualInspections*LSP	-0.02	0.9863

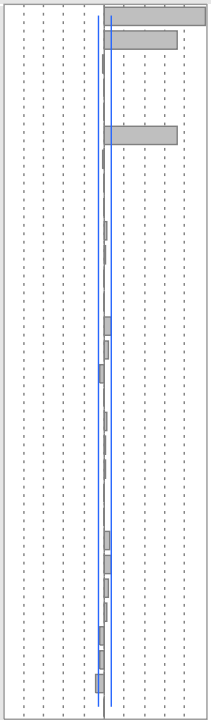


Figure I.3: Factors vs. LRU Repairs, Runway Visual Inspections

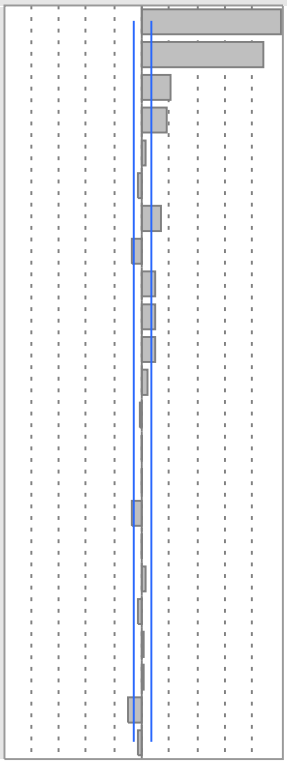
Term		Lenth t-Ratio	Individual p-Value
ParticleDiameter		26.37	<.0001*
DTAEngineInspections		23.36	<.0001*
LSP		5.69	<.0001*
ScanRate		4.91	<.0001*
RunwaySweeps		0.87	0.3815
PAutoDetection		-0.66	0.5115
ParticleDiameter*DTAEngineInspections		3.53	0.0004*
ParticleDiameter*LSP		-1.86	0.0637
DTAEngineInspections*LSP		2.51	0.0120*
ParticleDiameter*ScanRate		2.68	0.0073*
DTAEngineInspections*ScanRate		2.60	0.0096*
LSP*ScanRate		1.07	0.2869
ScanRate*ScanRate		-0.42	0.6783
ParticleDiameter*RunwaySweeps		0.12	0.9064
DTAEngineInspections*RunwaySweeps		0.10	0.9252
LSP*RunwaySweeps		-2.02	0.0444*
ScanRate*RunwaySweeps		-0.13	0.9003
ParticleDiameter*PAutoDetection		0.75	0.4579
DTAEngineInspections*PAutoDetection		-0.83	0.4069
LSP*PAutoDetection		0.28	0.7864
ScanRate*PAutoDetection		0.44	0.6626
RunwaySweeps*PAutoDetection		-2.71	0.0070*
PAutoDetection*PAutoDetection		-0.67	0.5091

Figure I.4: Factors vs. LCCs, Runway Automated Inspections

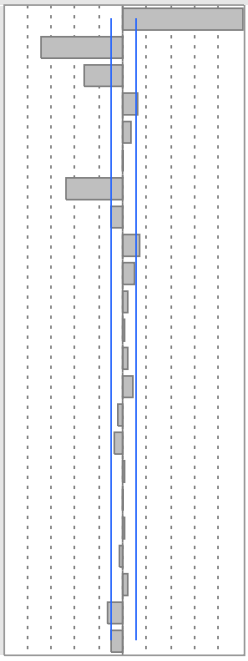
Term		Lenth t-Ratio	Individual p-Value
ParticleDiameter		15.70	<.0001*
DTAEngineInspections		-10.85	<.0001*
LSP		-5.09	<.0001*
ScanRate		2.06	0.0380*
RunwaySweeps		1.19	0.2451
PAutoDetection		-0.02	0.9850
ParticleDiameter*DTAEngineInspections		-7.36	<.0001*
ParticleDiameter*LSP		-1.50	0.1390
DTAEngineInspections*LSP		2.19	0.0277*
ParticleDiameter*ScanRate		1.44	0.1579
DTAEngineInspections*ScanRate		0.74	0.4625
LSP*ScanRate		0.27	0.7804
ScanRate*ScanRate		0.67	0.5080
ParticleDiameter*RunwaySweeps		1.36	0.1834
DTAEngineInspections*RunwaySweeps		-0.74	0.4642
LSP*RunwaySweeps		-1.16	0.2559
ScanRate*RunwaySweeps		0.29	0.7722
ParticleDiameter*PAutoDetection		0.09	0.9309
DTAEngineInspections*PAutoDetection		0.16	0.8690
LSP*PAutoDetection		-0.40	0.6895
ScanRate*PAutoDetection		0.73	0.4731
RunwaySweeps*PAutoDetection		-1.98	0.0472*
PAutoDetection*PAutoDetection		-1.56	0.1220

Figure I.5: Factors vs. LRU Replacements, Runway Auto. Inspections

Term	Lenth t-Ratio	Individual p-Value
DTAEngineInspections	43.02	<.0001*
ParticleDiameter	30.40	<.0001*
ScanRate	5.55	<.0001*
PAutoDetection	-1.58	0.1123
LSP	-0.61	0.5403
RunwaySweeps	0.59	0.5586
DTAEngineInspections*ParticleDiameter	30.40	<.0001*
DTAEngineInspections*ScanRate	5.55	<.0001*
ParticleDiameter*ScanRate	4.65	<.0001*
ScanRate*ScanRate	0.99	0.3210
DTAEngineInspections*PAutoDetection	-1.58	0.1123
ParticleDiameter*PAutoDetection	-0.53	0.5965
ScanRate*PAutoDetection	0.48	0.6297
PAutoDetection*PAutoDetection	0.07	0.9425
DTAEngineInspections*LSP	-0.61	0.5403
ParticleDiameter*LSP	2.12	0.0346*
ScanRate*LSP	0.56	0.5762
PAutoDetection*LSP	1.11	0.2626
DTAEngineInspections*RunwaySweeps	0.59	0.5586
ParticleDiameter*RunwaySweeps	-0.08	0.9281
ScanRate*RunwaySweeps	-1.11	0.2635
PAutoDetection*RunwaySweeps	-0.31	0.7556
LSP*RunwaySweeps	-0.45	0.6566

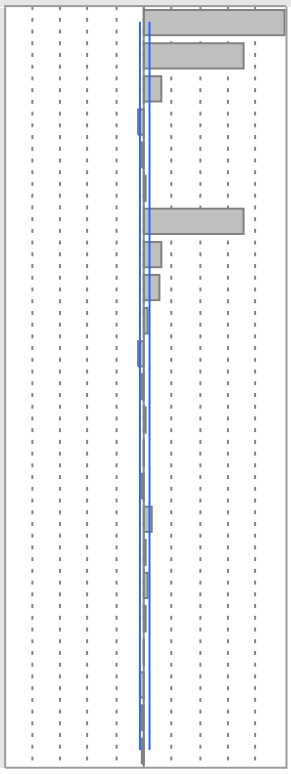


Figure I.6: Factors vs. LRU Repairs, Runway Auto. Inspections

REFERENCES

[1] Kelly, M., "THE 1992 CAMPAIGN: The Democrats - Clinton and Bush Compete to Be Champion of Change; Democrat Fights Perceptions of Bush Gain", The New York Times, October 31, 1992.

[2] INCOSE Affordability Working Group, outcome of presentation to INCOSE board at International Symposium, June 2011.

[3] NDIA Affordability Working Group, outcome for initial dissemination to NDIA, June 2011.

[4] Walker, D., "Trends in U.S. Military Spending." Jul 2013. Council on Foreign Relations. Feb 2014.

[5] Gates, R. M., "A Balanced Strategy - Reprogramming the Pentagon for a New Age", Foreign Affairs, January/February 2009.

[6] Shalal, A., "U.S. Army and suppliers brainstorm ways to do more with less", Reuters, Feb 19, 2014.

[7] Defense Budget Priorities and Choices—Fiscal Year 2014", Department of Defense, April 2013.

[8] F-35 Joint Strike Fighter: Current Outlook Is Improved, but Long-Term Affordability Is a Major Concern. GAO-13-309. Washington, D.C.: March 2013.

[9] Affordability and Life-Cycle Resource Estimates, Defense Acquisition University Defense Acquisition Guidebook, June 28, 2013.

[10] DoD Directive 5000.01, November 20, 2007.

[11] Weapon Systems Acquisition Reform Act of 2009 (WSARA), Public Law 111–23, May 22, 2009.

[12] Weapons Systems Acquisition Reform Act, Senate Armed Services Committee 2009.

[13] Hillen, J., “Defense's Death Spiral: The Increasing Irrelevance of More Spending”, Foreign Affairs, July/Aug 1999.

[14] Lorell, M. A., Kennedy M., R Leonard, R.S., Munson, K., Abramzon, S., An, D.L and Guffey, R.A., “Do Joint Fighter Programs Save Money?” RAND Corporation, 2013.

[15] Operating and Support Cost-Estimating Guide, Office of the Secretary of Defense, Cost Assessment and Program Evaluation (CAPE), March 2014.

[16] Cowles, B.A., “High cycle fatigue in aircraft gas turbines - an industry perspective”, International Journal of Fracture 80: 147-163, 1996.

[17] “Best Practices for the Mitigation and Control of Foreign Object Damage-Induced High Cycle Fatigue in Gas Turbine Engine Compression System Airfoils,” NATO Research and Technology Organization (RTO) TR-AVT-094, June 2005.

[18] Airline Industry Overview, MIT Global Airline Industry Program http://web.mit.edu/airlines/analysis/analysis_airline_industry.html

[19] Williams, G., “The Airline Industry and the Impact of Deregulation”. Aldershot, Hants, England: Avebury Aviation, 1994. Print.

[20] Winston, C., “Lessons from the U.S. Transport Deregulation Experience for Privatization”, Discussion Paper 2009-20 - © OECD/ITF, 2009.

[21] Pearce, B., “The outlook for commercial air transport”, IATA Economics Report, December 2013.

[22] 2011 Economic Report, Airlines for America, 2011.

[23] Current Market Outlook 2013 – 2032, Boeing, 2013.

[24] Pearce, B., “Airlines worldwide: The value they create and the challenges they face”, IATA Economics Report, July 2013.

[25] “Aviation Industry Performance: A Review of the Aviation Industry, 2008–2011”, Controlled Correspondence No. 2012-029, Office of Inspector General, U.S. Department of Transportation, September 24, 2012.

[26] International Energy Outlook 2013, U.S. Energy Information Administration, U.S. Department of Energy, July 2013.

[27] Annual Energy Outlook 2014 Early Release Overview, U.S. Energy Information Administration, U.S. Department of Energy.

[28] Dupuy, M.J.; Wesely, D.E.; Jenkins, C.S., "Airline fleet maintenance: Trade-off analysis of alternate aircraft maintenance approaches," Systems and Information Engineering Design Symposium (SIEDS), 2011 IEEE, April 29, 2011: pp. 29-34.

[29] McCartney, S., “How Airlines Spend Your Airfare”, The Wall Street Journal, June 6, 2012.

[30] McCreary, I., “Runway Safety: FOD, Birds, and the Case For Automated Scanning”, Insight SRI, 2010.

[31] “Financial Forecast: Profits Still Rising but so is Risk”, IATA economic briefing, March 2014.

[32] Our Consulting Philosophy, Jet Engine Consulting 2008. Source: http://www.jetengineconsulting.com/pdf/JEC_Consulting_Philosophy.pdf.

[33] FOD Prevention Guideline, National Aerospace FOD Prevention, Inc. (NAFPI), http://www.nafpi.com/nafpiguideline.pdf.

[34] Blanchard, B.S., Fabrycky, W.J., (Wolter J.), Systems Engineering and Analysis, 1981: pp. 24.

[35] Bobinis, J.S., “Meta Logistics: A System of Systems Exploration of Design for Operational Effectiveness”.

[36] FAA Fact Sheet – Foreign Object Debris (FOD), November 2013.

[37] “Did You Know That?”, Bird Strike Committee USA, http://www.birdstrike.org/.

[38] Scott, B., Aykent, M., “FOD Prevention - It Takes a Team”, 2007, presented NAFPI Conference 2010, http://www.nafpi.com/conference/2010/presentations/FODPreventionItTakesATeam.pdf.

[39] “On the Road to Zero Debris”, Frontiers, Boeing, Volume VIII, Issue IV, August2009.

[40] “FOE Quick-Start Guide: Everything You Need to Build Your Own FOD Prevention Program”, FOE Program guide for suppliers, Lockheed Martin Space Systems Company, 2013, http://www.lockheedmartin.com/content/dam/lockheed/data/corporate/documents/suppliers/SupplierFOEStartUp.doc.

[41] “Airport Foreign Object Debris”, U.S. Department of Transportation, Federal Aviation Administration (FAA), Advisory Circular No. 150/5210-24, September 30, 2010.

[42] “FOD is not just an airplane problem!: Tailoring FOE programs for manufacturing”, FOD elimination model, Lockheed Martin Missiles and Fire Control, presented NAFPI Conference 2011, http://www.nafpi.com/conference/2011/Presentations/FODisnotjustanairplaneproblem08-16-11.pdf.

[43] Wildlife Strikes to Civil Aircraft in the United States 1990–2012, U.S. Department of Transportation, Federal Aviation Administration, Federal Aviation Administration, National Wildlife Strike Database Serial Report Number 19, September 2013.

[44] “Foreign Object Damage/Foreign Object Debris (FOD) Prevention”, National Aerospace Standard (NAS412), Aerospace Industries Association of America, 2013.

[45] Carter, T.J., “Common Failures in Gas Turbine Blades”, Engineering Failure Analysis, 12th edition, 2005: pp. 237–247.

[46] Nicholas, T., “High Cycle Fatigue Management in Gas Turbine Engines”, NATO Research and Technology Organization (RTO), MP-17, October 1998.

[47] Chen, X., Hutchinson, J.W., “Foreign object damage and fatigue crack threshold: Cracking outside shallow indents”, International Journal of Fracture, 2001: volume 107, pp. 31–51.

[48] Dixon, M., “The Maintenance Costs of Aging Aircraft: Insights from Commercial Aviation”, RAND Corporation, 2006.

[49] Chaplin, G., Make It FOD Free, 2004.

[50] Procaccio, F. A., “Effectiveness of FOD Control Measures”, Master’s Thesis, Embry-Riddle Aeronautical University, October 2008.

[51] Crosby, P.B., Quality is Free: The art of Making Quality Certain, 1979.

[52] Mason, F.A., Kraus, D.C., Johnson, W.B., Watson, J., “Reducing Foreign Object Damage Through Improved Human Performance: Best Practices”, March 7, 2001.

[53] Human Factors, AMT Handbook addendum, 2011: chapter 14. http://www.faa.gov/regulations_policies/handbooks_manuals/aircraft/media/AMT_Handbook_Addendum_Human_Factors.pdf

[54] Rankin, W., MEDA Investigation Process for Human Factors, Boeing, 2003.

[55] “Contractor’s Flight and Ground Operations”, DCMA INST 8210.1C, August 2013.

[56] “Quality Management Systems - Requirements for Aviation, Space and Defense Organizations”, SAE Aerospace Standard (AS9100C), January 2009.

[57] “Quality Maintenance Systems - Aerospace - Requirements for Maintenance Organizations”, SAE Aerospace Standard (AS9110), January 2003.

[58] Provisional-ICAO, International Civil Aviation Organization (ICAO), 1944.

[59] Current Market Outlook 2013 – 2032, Boeing, 2013.

[60] Pearce, B., "Airlines worldwide: The value they create and the challenges they face", IATA Economics Report, July 2013.

[61] Lorell, M. A., Kennedy M., R Leonard, R.S., Munson, K., Abramzon, S., An, D.L and Guffey, R.A., "Do Joint Fighter Programs Save Money?" RAND Corporation, 2013.

[62] Beatty, D.N., Readdy, F., Gearhart, J.J., Duchatellier, R., "The Study of Foreign Object Damage Caused by Aircraft Operations on Unconventional and Bomb-Damaged Airfield Surfaces", June 1981.

[63] Zaklouta, H., "Cost of Quality Tradeoffs in Manufacturing Process and Inspection Strategy Selection", Master's Thesis, MIT, September 2011.

[64] Daniel, S.J., Reitsperger, W.D., "Linking Quality Strategy with Management Control Systems: Empirical Evidence from Japanese Industry", Accounting, Organizations and Society, 1991: Vol. 16, No. 7, pp. 601-618.

[65] Zeithaml, V.A., "Consumer perceptions of price, quality, and value: a means-end model and synthesis of evidence", Journal of Marketing, July 1988: Volume 52, pp. 2 – 22.

[66] "Controlling FOD In A Manufacturing Environment – Part 1", FODProgram.com, http://fodprogram.com/controling-fod-in-a-manufacturing-environment-part-1/.

[67] "Controlling FOD In A Manufacturing Environment – Part 2", FODProgram.com, http://fodprogram.com/controling-fod-in-a-manufacturing-environment-part-2/.

[68] "Controlling FOD In A Manufacturing Environment – Part 3", FODProgram.com, http://fodprogram.com/controling-fod-in-a-manufacturing-environment-part-3/.

[69] "Quality Management Systems - Requirements for Aviation, Space and Defense Organizations", SAE Aerospace Standard (AS9100C), January 2009.

[70] JO-2 Airfield Operations Manual, Flight Operations Aviation Management Office, Moffett Field. July 2009.

[71] Shingō, S., “Zero Quality Control: Source Inspection and the Poka-yoke System”, 1986.

[72] Fechushak, J. C., “Eliminated FOD by Raising the Bar and Changing a Culture”, presented at 2010 NAFPI Conference, 2010.

[73] Automated Foreign Object Debris (FOD) Detection System Evaluation, Federal Aviation Administration (FAA) Airport Technology Research and Development Branch, 2011. http://www.airporttech.tc.faa.gov/Design/fod.asp

[74] FOD technologies vendor, F.O.D. Control Corporation, http://www.fodcontrol.com/

[75] “Airport Foreign Object Debris (FOD) Detection Equipment”, U.S. Department of Transportation, Federal Aviation Administration (FAA), Advisory Circular No. 150/5220-24, September 30, 2009.

[76] Sylos Labini, P., Oligopoly and Technical Progress, 1969.

[77] Andersen, T.B., Asche, F., Hellen ROll, K., "Oligopoly and Oligopsony Power in Concentrated Supply Chains", April 2008.

[78] Zaklouta, H., “Cost of Quality Tradeoffs in Manufacturing Process and Inspection Strategy Selection”, Master’s Thesis, MIT, September 2011.

[79] Daniel, S.J., Reitsperger, W.D., “Linking Quality Strategy with Management Control Systems: Empirical Evidence from Japanese Industry”, Accounting, Organizations and Society, 1991: Vol. 16, No. 7, pp. 601-618.

[80] Carter, T.J., “Common Failures in Gas Turbine Blades”, Engineering Failure Analysis, 12th edition, 2005: pp. 237–247.

[81] Meher-Homji, C. B., "Gas Turbine Blade Failures - Causes, Avoidance, and Troubleshooting", Proceedings of the 27th Turbomachinery Symposium, 1998.

[82] Federal Aviation Administration (FAA) Part 25 - Airworthyness Standards.

[83] Peters, J.O., Roder, O., Boyce, B.L., Thompson, A.W. and Ritchie, R.O., “Role of Foreign-Object Damage on Thresholds for High-Cycle Fatigue in Ti-6Al-4V”, Metallurgical and Materials Transactions A, 1999.

[84] Boyce, B.L., Mehta, A., Peters, J.O., and Ritchie, R.O., “Synchrotron Mesodiffraction: A Tool for Understanding Turbine Foreign Object Damage”, Stanford Synchrotron Radiation Laboratory, 2003.

[85] Taguchi, G., Chowdhury, S., Wu, Y., Taguchi's Quality Engineering Handbook, 2005.

[86] Reason, J. “Human error: models and management”, BMJ Volume 320, pp 768-770, 2000.

[87] Nicholas, T., Barber, J., Bertke, R.S., "Impact Damage on Titanium Leading Edges from Small Hard Objects", AFML-TR-78-173, November 1978.

[88] Bertke, R.S., "Local Leading Edge Damage from Hard Particle and Soft Body Impacts", AFWAL-TR-82-2044, May 1982.

[89] Chen, X., Hutchinson, J.W., "Particle Impact on Metal Substrates with Application to Foreign Object Damage to Aircraft Engines", Journal of the Mechanics and Physics of Solids, 2002, Vol. 50, pp. 2669 - 2690.

[90] Thompson, S.R., Nicholas, T., Ruschau, J.J., Porter, W.J.,Buchanan, D.J., "An Investigation into Impacting Techniques for Simulating Foreign Object Damage and Their Influence on the Fatigue Limit Strength of Ti-6Al-4V",RTO-MP-AVT-121, October 2005.

[91] Hamrick, J.L., “Effects of Foreign Object Damage From Small Hard Particles on the High-Cycle Fatigue Life of Ti-6Al-4V”, September 1999.

[92] Peters, J.O., Boyce, B.L., Chen, X., McNaney, J.M., Hutchinson, J.W., Ritchie, R.O.,"On the application of the Kitagawa–Takahashi diagram to foreign-object damage and high-cycle fatigue", Engineering Fracture Mechanics, 2002, Vol. 69, pp. 1425–1446.

[93] Greenhalgh,E.S., Chichester,G. A. F., Mew, A.,Slade, M., "Characterisation of the Realistic Impact Threat from Runway Debris", The Aeronautical Journal, October 2001, pp.557.

[94] Krone, N.J., Beatty, D.N., "Aerodynamics Critical to the Operations of Tactical Fighters from Bomb Damaged Runways", AIAA-83-1861, AIAA Applied Aerodynamics Conference, July 1983.

[95] Pratt & Whitney F100-PW-229 Brochure, S16219-E.03.15.

[96] Murphy, J.P., *Intake Ground Vortex Aerodynamics*, Ph.D. thesis, Cranfield University, 2008. Print.

[97] Glenny, D.E., Pyestock, N.G.T.E., "Ingestion of Debris into Intakes by Vortex Action", C.P. No. 1114, 1970.

[98] Lee, A.S., Singh, R., Probert, S.D., "Modelling of the Performance of a F100-PW229 Equivalent Engine under Sea-level Static Conditions", AIAA 2009-5018, 2009.

[99] https://www.grc.nasa.gov/www/k-12/airplane/airprop.html

[100] Murphy, J.P., MacManus, D.G., “Inlet Ground Vortex Aerodynamics under Headwind Conditions”

[101] Calvert, J.B., "The Vortex", July 2003.

[102] Bolster, D., Hershberger, R.E., Donnelly, R.J., "Dynamic similarity, the dimensionless science", Physics Today, September 2011.

[103] Jischke, M.C., Parang, M., "On Tornado Funnels", Proc. Okla. Acad. Sci., 1978, volume 58, pp. 81-87.

[104] Rodert, L.A., Garrett, F.B., "Ingestion of Foreign Objects into Turbine Engines by Vortices", NACA TN 3330, 1955.

[105] Falck, N., Axial Flow Compressor Mean Line Design, Master thesis, Lund University, 2008. Print.

[106] Smith, L.S., 1-D Mean Line Code Technique to Calculate Stage-by-Stage Compressor Characteristics, 1999. Print.

[107] F100-220/220E Compressor Fan Blades PWA 46 D-GUN Plasma Application Solicitation Number: FA8100-14-R-0009.

[108] Koff, B.L., "F100-PW-229 Higher Thrust in Same Frame Size", Journal of Engineering for Gas Turbines and Power, APRIL 1989, Vol. 111, pp. 187.

[109] Walton, J.T., Burcham, F.W., "Augmentor Performance of an F100 Engine Model Derivative Engine in an F-15 Airplane", NASA Technical Memorandum 86745, 1986.

[110] Boyce, M.P. "Axial-Flow Compressors", Gas Turbine Engineering Handbook. Oxford: Elsevier Science & Technology, 2011. pp. 163-193. Print.

[111] Schulze, W.M., Erwin, J.R., Ashby, G.C., "NACA 65-Series Compressor Rotor Performance with Varying Annulus-Area Ratio, Solidity, Blade Angle, and Reynolds Number and Comparison with Cascade Results", NACA TV 4130, 1957.

[112] http://airfoiltools.com.

[113] Anderson, John D. *Modern Compressible Flow: With Historical Perspective*. McGraw-Hill, 1990. Print.

[114] Hill, Philip & Peterson, Carl. *Mechanics and Thermodynamics of Propulsion.* Addison-Wesley, 1992. Print.

[115] Glassman, A.J., "Blading Models for TURBAN and CSPAN Turbomachine Design Codes", NASA Contractor Report 191164, 1993.

[116] Wark, K., Thermodynamics, 4th ed. New York: McGraw-Hill, 1983. pp. 785–86.

[117] Verschuuren, G., “Monte Carlo Simulations in Excel”, video in YouTube. Source: https://www.youtube.com/watch?v=UeGncSFijUM.

[118] Fayer, J., A Modeling Study to predict the Residual Stresses caused by an Impact on a Ti64 Sample, Master thesis, Rensselaer Polytechnic Institute, 2010. Print.

[119] http://airfoiltools.com/plotter/index

[120] Bauchau, Olivier A., Craig, James I, *Structural Analysis to Aerospace Structures*. Springer, 2009. pp.184.

[121] Mecholsky, John J., Dental Materials, 1995. volume 11, pp. 111-112.

[122] Zehnder, Alan T., *Fracture Mechanics*. Springer, 2012.

[123] Nalla, R.k., Campbell, J.P., Ritchie, R.O., "Mixed-mode, high-cycle fatigue-crack growth thresholds in Ti–6Al–4V: Role of small cracks", International Journal of Fatigue, 2002. vol. 24, pp 1047–1062.

[124] Park, Ji Eun, "Ti-6Al-4V Durability Method Development and Test Verification on Small Crack", Aircraft Structural Integrity Program Conference, Presentation, 2014.

[125] Nalla, R.K.,, Boyce, B.L., Campbell, J.P., Peters, J.O., Ritchie, R.O., "Influence of Microstructure on High-Cycle Fatigue of Ti-6Al-4V: Bimodal vs. Lamellar Structures", Metallurgical and Materials Transactions, 2002. vol. 33a, pp 899–918.

[126] Broek, David. *The Practical Use of Fracture Mechanics* . Kluwer Academic Publishers, 1989.

[127] Garrison, B., et al., "High Cycle Fatigue (HCF) Science and Technology Program 1998 Annual Report", United States Air Force Technical Report, AFRL-PR-WP-TR-2001-2010, Wright-Patterson AFB, OH, January 2000.

[128] Hudak, S.J. and Davidson, D.L., “Characterization of Service Induced FOD,” United States Air Force Technical Report, Improved High Cycle Fatigue Life Prediction, Appendix 5A, AFRL-ML-WP-TR-2001-4159, Wright-Patterson AFB, OH, January 2002.

[129] Chen, D.L., Weiss, B., Stickler, R., "A New Approach for the Determination of Stress Intensity Factors for the Finite Width Plate", Engineering Fracture Mechanics, 1994. vol. 48, pp. 561-571.

[130] Sameezadeh, M., Farhangi, H., "Fracture Analysis of Generator Fan Blades", Applied Fracture Mechanics, 2012. pp 311-330.

[131] Ritchie, R.O., Davidson, D.L., Boyce, B.L., Campbell, J.P., Roder, O., "High-cycle Fatigue of Ti–6Al–4V", Fatigue Fracture Engineering Material Structures, 1999. vol. 22, pp. 621–631.

[132] Nalla, R.K., Altenberger, I., Noster, U., Liu, G.Y., Scholtes, B., Ritchie, R.O., "On the influence of mechanical surface treatments - deep rolling and laser shock peening - on the fatigue behavior of Ti-6Al-4V at ambient and elevated temperatures", Materials Science and Engineering, 2003. Vol. A355, pp. 216-230.

[133] Laser Peening Technical Bulletin, Concurrent Technologies Corporation, 2008. Source: http://www.nmc.ctc.com/useruploads/file/publications/LaserPeeningTechBulletin.pdf.

[134] Brajer, J., "Application of Laser Shock Peening", HiLASE and DPSSLasers projects at Czech Technical University, 2014.

[135] Prevey, P.S., Shepard, M.J., Smith, P.R., "The Effect of Low Plasticity Burnishing (LPB) on the HCF Performance and FOD Resistance of Ti-6Al-4V", Proceedings: 6th National Turbine Engine High Cycle Fatigue (HCF) Conference, 2001.

[136] Niku-Larl, A., "An Overview of Shot-Peening", International Conference on Shot Peening and Blast Cleaning, 1996.

[137] Corran, R.S.J, Williams, S.J., "Lifing methods and safety criteria in aero gas turbines", Engineering Failure Analysis, 2007. Vol. 14, pp. 518–528.

[138] Turbine and Compressor Rotors Type Certification Substantiation Procedures, FAA AC 33-3. 1968.

[139] Fallenstein, J., "F100-PW-220/E/229 Engine Life Management Plan and Part Lifing Programs", Pratt & Whitney, 8th Israeli Symposium on Jet Engines and Gas Turbines, 2009. Presentation. ASC Public Release Approval # 88ABW-2009-4757 dated 11/12/09.

[140] Safety Investigation Report, Hellenic Air Force F-16D BLK 50, S/N 93-1084 341SQ/111CW, Albacete AFB (Spain), Albacete. JANUARY 26, 2015.

[141] Foster, R., Mellier, R., "Built to Last!" McDonnell Douglas Aerospace Digest, 1994. Vol 41 No. 2, pp. 11.

[142] Singh, G., Grandhi, R.V., Stargel, D.S., "Modeling and Parameter Design of a Laser Shock Peening Process", International Journal for Computational Methods in Engineering Science and Mechanics, 2011.

[143] 2015 Index of U.S. Military Strength, The Heritage Foundation, 2014.

[144] Total Car Operating Costs 2015, Automotive Fleet, 2014.

[145] Eliminating Major Gaps in DoD Data on the Fully-Burdened and Life-Cycle Cost of Military Personnel: Cost Elements Should be Mandated by Policy. Final Report to the Secretary of Defense, Reserve Forces Policy Board, 2013.

[146] Johnston, Katie, "Logan Airport installs runway scanning system", Boston Globe. November 16, 2013.

[147] "Stratech's iFerret™ Wins in Hong Kong", Stratech Systems Limited. Press release. February 24, 2015. http://www.thestratechgroup.com/news/p_news145.html.

[148] SENTINEL FOD Detection Performance Evaluation Trial, TR-01-2014, Pharovision, June 2014.

[149] Shukla, P., Swanson, P.T., Page, C.J., "Laser Shock Peening and Mechanical Shot Peening Processes Applicable for the Surface Treatment of Technical Grade Ceramics: A Review", Journal of Engineering Manufacture, Proceedings of The Institution of Mechanical Engineers, Part B, 2013.

[150] Srinivasan, A.V., "Flutter and Resonant Vibration Characteristics of Engine Blades", American Society of Mechanical Engineers, International Gas Turbine & Aeroengine Congress & Exhibition, 1997.

[151] Engine Health Management, Rolls Royce. http://www.rolls-royce.com/about/our-technology/enabling-technologies/engine-health-management.aspx#sense.

[152] The Risk Analysis Correspondence Course, FAA Engine and Propeller Directorate's risk analysis process, FAA.

[153] Meadows, T.A., *Analysis of F/A-18 Engine Maintenance Costs Using the Boeing Dependability Cost Model*, Master thesis, Naval Postgraduate School, 1994. Print.

[154] Vanden Brook, Tom. "Harsh conditions are foiling Russian jets in Syria", USA Today, October 25, 2015.

[155] Younossi, O., Arena, M.V., Moore, R.M., Lorell, M., Mason, J., Graser, J.C., "Military Jet Engine Acquisition: Technology Basics and Cost-Estimating Methodology", RAND Corporation, 2002.